Mäenpää et al. EURASIP Journal on Wireless Communications and Networking 2012, 2012:121
http://jwcn.eurasipjournals.com/content/2012/1/121

EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Using RELOAD and CoAP for wide area sensor and actuator networking

Jouni Mäenpää*, Jaime Jiménez Bolonio and Salvatore Loreto

Abstract

In this article, we propose a new architecture for wide area sensor and actuator networking. The architecture is based on combining two protocols being standardized by the Internet Engineering Task Force, REsource LOcation And Discovery (RELOAD) and Constrained Application Protocol (CoAP). To integrate CoAP and RELOAD, we introduce a CoAP application usage for RELOAD. The architecture provides a decentralized peer-to-peer rendezvous service for CoAP nodes in Wireless Sensor Networks (WSNs). Our architecture also enables a peer-to-peer federation of geographically distributed WSNs. This is supported by proxy nodes that are part of the WSN but also connect to a RELOAD overlay network via cellular Internet access. Due to the use of RELOAD, the system does not need to rely on centralized services such as DNS service discovery (DNS-SD) or central resource directories to discover sensors and resources. Other features of the architecture include integration to web, self-organization, scalability, and robustness. We evaluate the proposed architecture through simulations and real-life measurements, and compare its performance to a traditional client/server architecture.

Keywords: sensor and actuator networking, Constrained Application Protocol, REsource LOcation And Discovery, Internet of Things

1 Introduction

The rapid increase in the number of IP-enabled embedded devices is giving rise to the Internet of Things. The vision behind the Internet of Things is that everyday objects (e.g., sensors, actuators, consumer electronics, and industrial devices) become interconnected, IP-addressable, and an integral part of the services in the Internet. According to some visions, the number of connected devices will grow to more than 50 billion in the next ten years [1]. Further, if considering not only devices but all kinds of objects, we may see even trillions of connected things being added to the Internet [2]. Thus, there is a need for network architectures that can support the exponential growth in the number of devices. The solutions are likely to benefit from being open and standards-based to support interoperability between various ecosystems and vertical industries.

One main factor enabling the Internet of Things is the cost of wireless modules. Technological advances, broad deployment, and economies of scale of third Generation partnership project (3GPP) standards make them affordable and attractive for many applications and competitive compared to other technologies, especially when there is a need for wide area connectivity. It is foreseen that the geographical coverage and flexibility of cellular mobile connections will make them the dominant technology for connecting things to the Internet [3].

Although cellular technologies are attractive for wide area sensor and actuator networking, constrained wireless sensor networks (WSNs) such as LoW-power wireless Personal Area Networks (LoWPANs) are better served by short-range radio technologies like IEEE 802.15.4, which specifies the physical and media access control layer for low-rate WSNs.

In this article, we propose a new architecture for wide area sensor and actuator networking. The architecture provides a peer-to-peer (P2P) federation of geographically distributed WSN islands. Nodes equipped with both cellular and WSN radio interfaces sit at the edge of the WSNs and participate in a P2P overlay network that provides a common namespace, rendezvous, and other services. The solution is based on emerging open Internet standards such as Constrained Application Protocol (CoAP) [4] and REsource LOcation And Discovery

* Correspondence: jouni.maenpaa@ericsson.com
Ericsson Research, Hirsalantie 11, Jorvas 02420, Finland

(RELOAD) [5], both of which are currently being specified by the Internet Engineering Task Force (IETF).

The remainder of the article is structured as follows. Section 2 introduces the main technologies that our architecture utilizes. Section 3 presents related study. Section 4 lists the requirements of the use cases for which our architecture is targeted. Section 5 describes the architecture. Section 6 presents the CoAP usage for RELOAD. Section 7 describes the setup of the simulations that we used to evaluate the architecture. Section 8 presents the results of the simulations and our real-life measurements. Section 9 concludes the article.

2 Overview of main technologies

2.1 Constrained Application Protocol (CoAP)

Constrained Application Protocol (CoAP) [4] is a specialized web transfer protocol. It realizes the Representational State Transfer (REST) architecture for the most constrained nodes. CoAP can be used not only between nodes on the same constrained network but also between constrained nodes and nodes on the Internet. Nodes on the Internet that do not support CoAP but only the Hypertext Transfer Protocol (HTTP) can talk to CoAP nodes and vice versa since CoAP can be translated to HTTP for integration with the web. CoAP can also be used between devices in different constrained networks interconnected by an internet, which makes it very suitable for our architecture that federates separate constrained networks. Application areas of CoAP include different forms of machine-to-machine communication. CoAP provides a request/response interaction model between application endpoints, supports built-in resource discovery, and includes key web concepts such as Uniform Resource Identifiers (URIs) and content-types. CoAP uses unreliable datagram-oriented transport (i.e., UDP). CoAP meets the specialized requirements of constrained environments such as low overhead, simplicity, and ability to deal with sleeping nodes. The main example of operating environments CoAP targets is 6LoWPANs (IPv6 over LoW Power wireless Area Networks). However, CoAP also operates over traditional IP networks. CoAP has been extended to enable clients to establish observation relationships between themselves and resources [6]. To discover sensors and resources, CoAP may use either web linking [7], central resource directory [8], or DNS service discovery (DNS-SD) [9]. Web linking is not practical in many scenarios due to sleeping nodes, disperse networks, or networks where multicast traffic is inefficient. Further, web linking does not solve the problem of discovering the IP address of the CoAP server. Resource directories and DNS-SD require support from centralized entities. CoAP is neutral with respect to use of DNS. At the time of writing this article, the use of DNS and DNS-SD had not yet been defined for CoAP. In this article, we will describe how a RELOAD overlay network can be used as an alternative to centralized services such as DNS-SD and resource directories.

2.2 IPv6 over LoW-power wireless Personal Area Networks (6LoWPAN)

A LoW-power wireless Personal Area Network (LoWPAN) is a simple low-cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements. Devices in a LoWPAN conform to the IEEE 802.15.4 standard. IEEE 802.15.4 devices typically have short range, low bit rate, low power, and low cost. Further, the devices are limited in their computational power, memory, and/or energy availability. The IPv6 over LoWPAN (6LoWPAN) working group of the IETF has defined encapsulation and header compression mechanisms that allow IPv6 packets to be sent and received over LoWPANs [10]. 6LoWPAN networks often have high packet error rates and a typical throughput of tens of kilobits per second [11].

2.3 REsource LOcation And Discovery (RELOAD)

REsource LOcation And Discovery (RELOAD) [5] is a P2P signaling protocol that is being specified by the Peer-to-Peer Session Initiation Protocol (P2PSIP) working group of the IETF. RELOAD provides a generic, self-organizing P2P overlay network service. Nodes can use the RELOAD overlay to route messages to other nodes and to store and retrieve data. RELOAD uses the Chord Distributed Hash Table (DHT) algorithm [12] as the default algorithm to organize the overlay. For Network Address Translator (NAT) traversal, RELOAD uses Interactive Connectivity Establishment (ICE) [13]. RELOAD supports two types of nodes: peers and clients, both of which are identified by node-IDs. Peers are nodes that run the DHT algorithm, route messages, and store data on behalf of other nodes. Clients are nodes that do not run the DHT algorithm and do not provide message routing and storage services. Instead, they use the services of the overlay by connecting to a peer. The data stored in a RELOAD overlay are referred to as resources, which are identified by resource-IDs.

New applications can utilize RELOAD by defining new usages. Our architecture makes use of a new CoAP usage for RELOAD. This usage allows a RELOAD overlay to be used as a distributed rendezvous, storage, and NAT traversal service for CoAP endpoints. The usage is similar to the SIP usage for RELOAD [14]. The CoAP usage for RELOAD will be described in more detail in Section 6.

2.4 Interactive Connectivity Establishment (ICE)

As discussed above, RELOAD uses ICE as a technique for NAT traversal. ICE uses the Session Traversal Utilities for NAT (STUN) [15] protocol and its extension, Traversal Using Relays around NAT (TURN) [16]. STUN is used by a host to determine the IP address and port allocated to it by a NAT, to test connectivity between two hosts, and as a keepalive protocol to maintain NAT bindings. TURN is used in situations when two hosts are unable to communicate without the help of a relay. TURN allows a host to control the relay and to exchange packets with its peers using the relay. The main steps in an ICE negotiation between two nodes include gathering of ICE candidates and connectivity checks. ICE candidates are transport addresses (i.e., IP address and port pairs) that can be potentially used to communicate with a node. Connectivity checks are used to test connectivity between ICE candidates of two nodes.

3 Related study

To the best of the authors' knowledge, the integration of CoAP and RELOAD for federating geographically distributed WSNs has not been proposed earlier. Previous research has focused on using DHTs within WSNs, including solutions like ScatterPastry [17], Virtual Cord Protocol (VCP) [18], Geographic Hash Table (GHT) [19], and Chord for Sensor Networks (CSN) [20]. The difference between previous research and our architecture is that our study does not use P2P technologies for communication within a single WSN but rather for interconnecting separate WSNs.

We have previously studied the performance of RELOAD in wireless networks in [21]. Further, we have studied the performance of ICE-based NAT traversal for P2P overlays in [22]. The difference between our earlier study and the present study is that in this article, we focus on large-scale cellular-only RELOAD overlay networks and on the CoAP usage for RELOAD. Whereas our previous study focused on the use of DHTs for interpersonal communication, the present study focuses on the use of DHTs for interconnection of WSNs.

4 Requirements

In this section, we will describe the requirements of the use cases for which our architecture is targeted. First, the devices interconnected by the architecture require wide area geographical coverage. For this, they use cellular mobile connections. Examples include devices equipped with Second Generation (2G) or Third Generation (3G) modules or gateway nodes with dual interfaces: a short-range WSN radio interface and a 2G or 3G interface. The devices are deployed in the wide area; distances between the devices are on the order of hundreds of meters or kilometers.

The systems require self-* properties, that is, they need to be self-configuring, self-organizing, self-optimizing, self-adjusting, and self-reliant. As an example, the system may use data from sensors to trigger decisions in actuators in an autonomous manner without the involvement of central application servers. New devices can be added to the system in a plug-and-play (zero configuration) fashion. The system should also have low capital expenditures (capex; that is, low investment in central servers and data centers) and operating expenditures (opex; e.g., low involvement from maintenance personnel).

The system should be usable both in simple use cases and use cases in which data from the devices is real-time and the total volume of the data high. Further, there may be a high number of devices. Thus, the system should be scalable; even if a large number of new nodes are added, this should not require investment in new capacity in central servers or data centers. Since cellular networks may assign private IPv4 addresses to the devices [23], the system should support NAT traversal and data relaying between NATed endpoints in a scalable manner. Due to the above-mentioned factors, the system benefits from a great degree of decentralization to reduce the load of or eliminate the need for central components.

The devices interconnected by the architecture are heterogeneous. Some of the devices are very constrained sensors utilizing WSN radio technologies, whereas other devices use also cellular technologies and have at least a moderate amount of central processing unit (CPU) power and random access memory (RAM). There are also nodes having both a WSN radio interface and a cellular interface.

The devices, even the most constrained sensors are assumed to typically have IP connectivity, which is provided by 6LoWPAN. However, non-IP based WSNs (e.g., ZigBee) can be integrated to the system through gateway nodes. As the final requirement, the resources hosted on the devices should be accessible from web applications.

5 Architecture

Our wide area sensor and actuator networking architecture that integrates the CoAP and RELOAD protocols is illustrated in Figures 1 and 2. In Figure 1, all of the nodes (i.e., sensors and actuators) are geographically distributed and thus use only cellular technologies for communication. We call such nodes Wide area Nodes (WNs). WNs act as peers or clients in a RELOAD overlay network. In constrast to Figure 1, in Figure 2, there

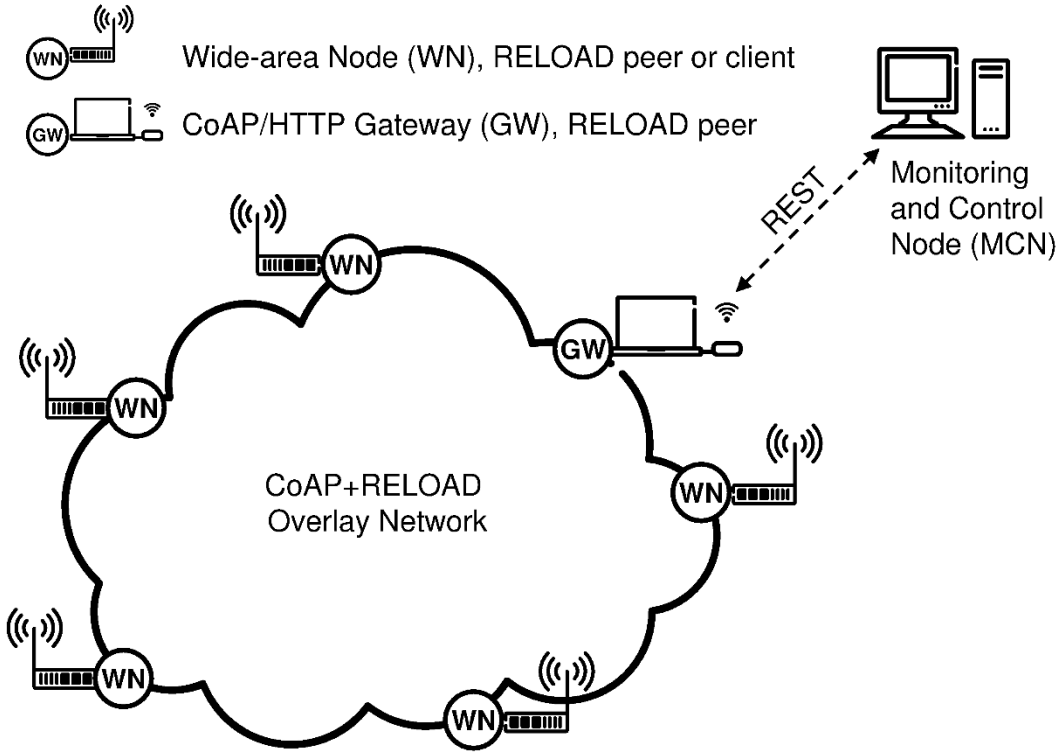

Figure 1 Architecture, all nodes are wide area nodes. The figure illustrates our architecture for using CoAP and RELOAD for wide area sensor and actuator networking in a scenario in which all nodes are Wide area Nodes (WNs).

are both geographically distributed nodes (i.e., WNs) and nodes located in a WSN in the same geographical area. The latter type of nodes are called Local Nodes (LNs). The WSN to which the LNs belong may be for instance a 6LoWPAN or a ZigBee network. In addition to LNs and WNs, Figure 2 also has nodes called Proxy Nodes (PNs), which, like WNs, are part of the RELOAD overlay. In addition to having a cellular interface, a PN also has an IEEE 802.15.4 radio interface. Since PNs are part of both a WSN and a RELOAD overlay, they can connect LNs in a WSN to the RELOAD overlay. Other types of nodes in Figures 1 and 2 include Gateway Nodes (GWs), and Monitoring and Control Nodes (MCNs). We will go through the different types of nodes in more detail in the sections below.

5.1 Local nodes

Local Nodes (LNs) are constrained devices such as sensors and actuators with limited resources (CPU, RAM, battery, etc.). There are two types of LNs: regular LNs and legacy LNs. This section describes regular LNs (henceforth, simply referred to as LNs), whereas legacy LNs are described in the subsection below. LNs are part of a 6LoWPAN and thus have IPv6 connectivity. The LNs act as CoAP endpoints. They host one or more resources that need to be discoverable in the RELOAD overlay. However, since LNs are constrained, and since 6LoWPAN application throughput may be low and payload size small, LNs are assumed not to be capable of acting as RELOAD clients. In other words, the LNs cannot use RELOAD directly to register their resources in the overlay. Instead, the PN takes care of registering the resources hosted by the LNs in the RELOAD overlay. All the application-level signaling between the LNs and PNs utilizes CoAP, as we will see in the examples of Section 6.

5.2 Legacy LNs

Also non-IP connected (i.e., legacy) sensors and actuators such as devices using ZigBee can become addressable in the RELOAD overlay network. We call such nodes legacy LNs. In a similar manner as with regular

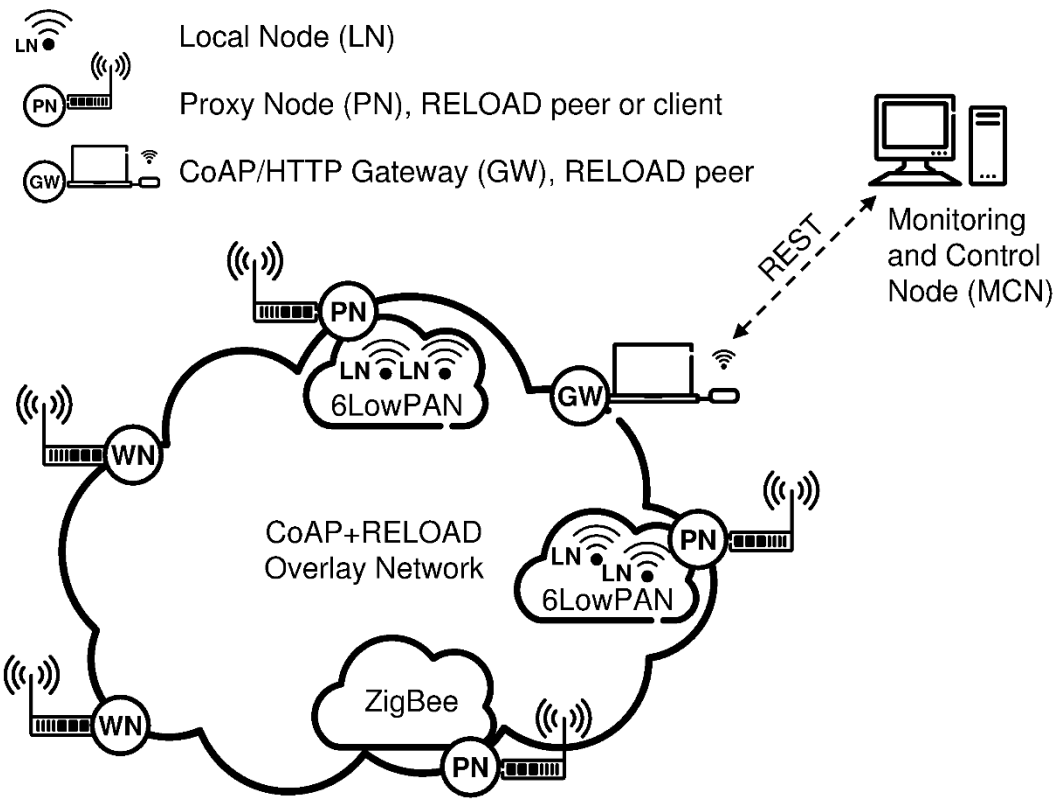

Figure 2 Architecture containing interconnected WSNs. The figure illustrates our architecture for using CoAP and RELOAD for wide area sensor and actuator networking in a scenario in which there are also Local Nodes (LNs) in addition to Wide area Nodes (WNs).

LNs, also in the case of legacy LNs a PN takes care of registering the resources hosted by the legacy LNs in the RELOAD overlay. The difference is of course that the signaling on the WSN side does not utilize CoAP/UDP/IP. In this scenario, the PN, having discovered legacy LNs in the WSN, assigns each legacy LN a RELOAD node-ID, and for each resource hosted by the legacy LN, a CoAP URI and a RELOAD resource-ID. The PN maintains a mapping between the WSN-specific IDs of the legacy LNs and the RELOAD node-IDs, and between the WSN-specific resource-IDs and CoAP URIs. When receiving CoAP messages addressed to the legacy LNs, the PN performs translation from CoAP/UDP/IP to the WSN-specific protocol stack (e.g., ZigBee protocol stack), thus acting as a gateway.

5.3 Wide area nodes

Wide area Nodes (WNs) are devices using a cellular technology such as 2G or 3G to connect to the Internet. WNs do not belong to a WSN and do not have a short-range radio (e.g., IEEE 802.15.4) interface. WNs typically participate as peers in the RELOAD overlay network but may in some cases alternatively act as clients (e.g., in order to limit resource consumption). When WNs act as RELOAD peers, they run the DHT algorithm and provide message routing and storage services to other peers in the overlay. Like LNs, also the WNs act as CoAP endpoints and host resources that can be accessed using CoAP.

5.4 Proxy nodes

Proxy Nodes (PNs) are located at the edge of a WSN. In the rest of this section, we will assume that the WSN is a 6LoWPAN, although legacy LNs can also be supported as described in Section 5.2. The PNs act as gateways between the 6LoWPAN and the Internet. Each 6LoWPAN is assumed to have its own domain in CoAP URIs to distinguish it from other 6LoWPANs. PNs use IEEE 802.15.4 on the 6LoWPAN side and connect to the Internet using a cellular radio interface. Like WNs, also PNs act as RELOAD peers or clients and thus participate in the RELOAD overlay. The difference between

WNs and PNs is that since PNs have an interface towards a 6LoWPAN, they can connect LNs to the RELOAD overlay network. Since a PN sits at the boundary of the 6LoWPAN and the Internet, it acts as a 6LoWPAN border router (6LBR) [24]. The PNs also act as CoAP endpoints and as CoAP proxies for CoAP nodes in the 6LoWPAN.

5.5 Gateway nodes

A Gateway Node (GW) acts as a peer in the RELOAD overlay network. In addition, the GW acts as a HTTP/CoAP proxy [25]. The purpose of a HTTP/CoAP proxy is to provide interoperability between HTTP and CoAP. Thus, as a HTTP/CoAP proxy, the GW provides web applications access to the resources in the WSNs interconnected by the RELOAD overlay network. It also enables CoAP clients to access resources on web servers.

5.6 Monitoring and control nodes

Monitoring and Control Nodes (MCNs) are HTTP clients that use the CoAP REST interface to access resources in the WSNs federated by the RELOAD overlay network. A GW node acts, as a HTTP/CoAP proxy, as the first point of contact for MCNs in the RELOAD overlay network.

The messaging between MCNs, GWs, PNs, and LNs will be described in Section 6.

5.7 Benefits of the architecture

Our architecture has several advantages. First, the architecture can be used to federate geographically distributed WSN islands. CoAP is used to provide a common namespace for resources in all interconnected WSNs. A sensor in one WSN can access the resources of a sensor in another WSN since the WSNs are interconnected by a RELOAD overlay.

The architecture enables decentralized wide-area sensor and actuator networks. RELOAD is used to provide a lookup, storage, message routing, and NAT traversal service. The RELOAD overlay maps CoAP URIs to contact information of sensors. In particular, since RELOAD is used, the system does not depend on services relying on central servers like resource directories or DNS-based service discovery (DNS-SD) [26] that is used by CoAP usages like the one specified in [9].

The architecture also integrates WSNs to the web. Due to the use of CoAP and GW nodes, web applications can access resources in the WSNs federated by the RELOAD overlay. CoAP clients can also access resources on web servers. In general, web integration of objects enables interesting new applications such as the Web of Things [27].

Due to the use of P2P technologies, the architecture is self-configuring, scalable, robust, and cost-efficient. It is scalable since each new PN added to the system brings extra resources to it. In contrast, in a client/server system, each new PN consumes additional resources on the central servers and thus eventually more capacity needs to be added, as we will show in Section 8. The system is also scalable when it comes to NAT traversal since in a P2P architecture, the peers can act as STUN and TURN servers to each other. In contrast, in a client/server system, centralized TURN servers are necessary. In a real-world system, a subset of the nodes will typically be behind the most restrictive types of NATs. When two such nodes need to communicate with each other, all the traffic between them needs to be relayed by a TURN server. If the volume of data is high and the system large, the relay servers need to have high capacity. However, in a P2P system, the responsibility for relaying data can be distributed among publically reachable peers. The architecture is robust since it is not dependent on centralized elements (i.e., central points of failure) for rendezvous and relaying of data. The system is cost-efficient since it has both low capex and perhaps more importantly, low opex.

6 CoAP usage for RELOAD

This section describes the CoAP usage for RELOAD proposed in this article. The usage provides four basic functions: registration of CoAP resources in the RELOAD overlay, rendezvous using dedicated ICE-negotiated connections for CoAP, rendezvous through encapsulating (i.e., tunneling) CoAP messages in the payload of RELOAD messages, and use of the RELOAD overlay as a cache for sensor data. In the examples of the section, we assume that all LNs are located in 6LoWPANs.

6.1 Registration of CoAP resources

From the viewpoint of LNs, a PN looks like a regular CoAP Server Discovery Server (CSDS). As defined in [28], a CSDS is a CoAP server which interacts with other CoAP servers, collects resource discovery information from them, and integrates the information into a resource directory. In the CoAP usage for RELOAD, also the CSDS functionality is distributed among the PNs and WNs participating in the RELOAD overlay. In other words, A PN uses the CoAP usage for RELOAD to register the resources of LNs in the distributed CSDS implemented by the RELOAD overlay.

The registration procedure that an LN that has been added to a 6LoWPAN uses to register its resourcecs in the RELOAD overlay is illustrated in Figure 3. In Step 1, the LN sends a CoAP POST request to the default

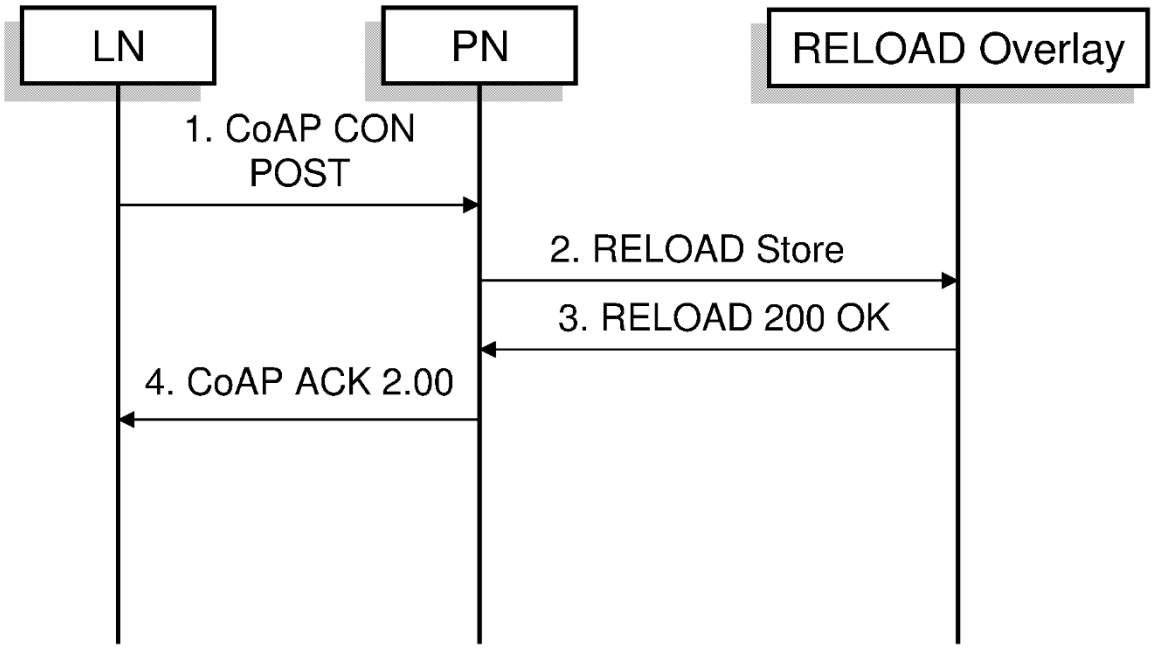

Figure 3 LN registering to RELOAD overlay. The figure illustrates the procedure that a Local Node (LN) uses to register its resources in the RELOAD overlay network through communicating with a proxy node (PN).

discovery URI of the PN. An LN discovers the IP address of the PN in the same way it discovers the IP address of a CSDS in [28], that is, using for instance static configuration, 6LoWPAN neighbor discovery options, or Dynamic Host Configuration Protocol for IPv6 (DHCPv6) options. The POST is either empty or contains a CoRE link format document [7] (i.e., a document describing the resources hosted by a node). If the request is empty, the PN will perform CoAP GET requests to learn the resources that the LN provides. If the request contains a link format document, the resources being registered are indicated directly in the document. In this example, we assume that the POST request contains a link format document indicating only one resource. For this resource, the PN needs to store a ⟨*hash(CoAP-URI), destination list*⟩ mapping into the overlay. *hash(X)* is a SHA-1 hash over *X* and *destination list* is a RELOAD destination list [5] specifying the Node-IDs through which the messages destined to the LN need to be routed. We use SHA-1 since RELOAD mandates it as the algorithm to compute resource-IDs. In the case of the CoAP usage for RELOAD, the destination list contains two entries: the node-ID of the LN and the node-ID of the PN behind which the LN is located. In Step 2, the PN uses the URI of the resource to create a RELOAD resource-ID for the resource record that is to be stored in the overlay by calculating a SHA-1 hash over the URI. The PN also creates a RELOAD node-ID for the LN and maintains a mapping between the node-ID and the node's transport address. Since the node-ID is created by the PN, the LN does not need to be aware of it. We assume no special relationship between the LN's node-ID and the PN's node-ID, such as the PN being responsible for the LN's node-ID in the overlay.

After the PN has created the node-ID, resource-ID, and destination list, it consults its local routing table and sends a RELOAD Store request to the appropriate next hop peer in the routing table. The Store is routed across the overlay to the peer responsible for the resource-ID. In Step 3, a RELOAD 200 OK reply to the Store is routed back to the PN. In Step 4, the PN sends a CoAP ACK with an immediate response back to the LN. Note that RELOAD hop-by-hop ACK messages have been omitted from the figure for brevity. The CoAP URIs that are used as RELOAD resource names and are thus hashed into RELOAD resource-IDs should be of the form defined in the CoAP specification [4] with the following additional recommendations. The URIs should include only registered names in the host part of the URI, and should not include port and query parts or trailing slashes.

Therefore, a typical RELOAD resource name would look as follows: *coap://sensor.example.net/temperature*. If there are situations in which it is necessary to have multiple alternative resource names (i.e., CoAP URIs)

for the same resource, one can store additional records of the type ⟨key = alternative - key, value = primary - key⟩ in the overlay. In such a record, the Resource-ID *alternative-key* has been created by calculating a SHA-1 hash over an alternative CoAP URI. The value part of the record, *alternative-key*, is the Resource-ID calculated using the primary CoAP URI. This way a party performing a lookup using an alternative CoAP URI can locate the actual resource stored using the primary CoAP URI.

6.2 Rendezvous using dedicated connections for CoAP

When the CoAP notifications between an observing node and a node hosting the observed CoAP resource are frequent enough to justify the use of dedicated connections for CoAP, the PNs of the communicating nodes use the CoAP usage for RELOAD to establish an ICE-negotiated UDP connection between themselves.

On the cellular network side, a PN may be located behind a NAT. Therefore, all the connections to and from the PN, including those established for communication between an LN and the outside world, need to be negotiated using ICE. Figure 4 shows the messaging associated with establishing a CoAP observation relationship between two LNs in different 6LoWPANs, including the ICE negotiation phase. In the figure, LN-A in 6LoWPAN-A starts observing a resource hosted by LN-B in 6LoWPAN-B. In Step 1, LN-A sends a CoAP GET to PN-A. The GET contains the request URI of the observed resource in a CoAP Proxy-Uri option. PN-A inspects the domain in the request URI. Since the domain is different from the domain of 6LoWPAN-A, PN-A performs in Step 2 a RELOAD lookup operation (Fetch request) in the RELOAD overlay to learn the destination list of LN-B. The Fetch request is routed across the overlay to the peer responsible for the resource-ID. The destination list is returned in Step 3 inside a RELOAD Fetch 200 OK response. In Step 4, PN-A sends a RELOAD Attach request across the overlay

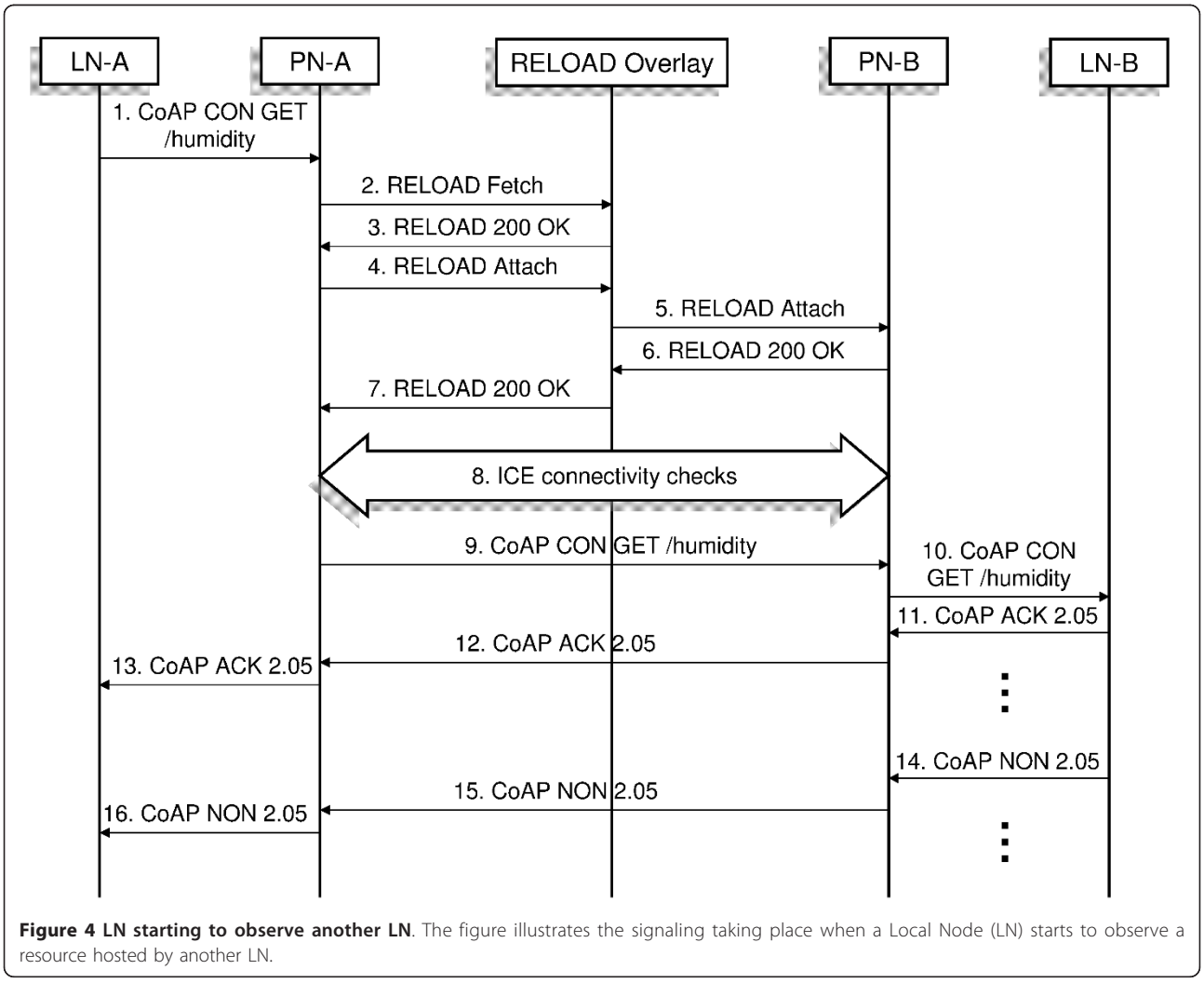

Figure 4 LN starting to observe another LN. The figure illustrates the signaling taking place when a Local Node (LN) starts to observe a resource hosted by another LN.

towards the first node (the node-ID of PN-B) on the destination list. The purpose of an Attach request is to set up a direct ICE-negotiated connection. The Attach carries PN-A's ICE candidate addresses. The Attach is routed to PN-B. In Steps 6 and 7, a RELOAD Attach 200 OK response with PN-B's ICE candidates is routed back to PN-A. In Step 8, PN-A and PN-B perform ICE connectivity checks to establish a direct UDP connection for CoAP. Once the connectivity checks are over, PN-A sends the CoAP GET to PN-B in Step 9 on the ICE-negotiated connection. PN-A is unaware that PN-B is a CoAP reverse proxy (a CoAP reverse proxy [4] is a proxy that receives requests as if it was the origin server for the target resource). In Step 10, PN-B, acting as a CoAP reverse proxy, sends the CoAP GET to LN-B. The GET establishes an observation relationship [6] between LN-A and the resource on LN-B. In Steps 11-13, a CoAP ACK carrying an immediate response is sent from LN-B to LN-A. In the future, whenever the state of the resource on LN-B changes, LN-B will notify LN-A by sending a new CoAP response. This is illustrated in Steps 14-16, in which a non-confirmable (NON) CoAP response using the 2.05 (Content) response code is assumed.

6.3 Rendezvous using tunneling

In some cases, such as when the frequency of CoAP notifications associated with an observation relationship is very low, it may be too expensive to establish dedicated ICE-negotiated connections for CoAP. In such cases, CoAP messages can be sent tunneled (i.e., encapsulated in the payload of RELOAD messages) across the overlay. For this, the CoAP usage for RELOAD defines a new RELOAD request which we call the Tunnel request. The use of Tunnel requests is illustrated in Figure 5. In the figure, LN-A in 6LoWPAN-A wants to access a resource hosted by LN-B in 6LoWPAN-B without establishing an observation relationship. Steps 1-3 in the figure are identical to Figure 4. In Step 4, having fetched the destination list of LN-B from the overlay, PN-A, instead of establishing an ICE-negotiated connection to PN-B, places the CoAP GET message in the payload of a RELOAD Tunnel request and routes the message towards PN-B, which is the first entry on the destination list. Having received the Tunnel request in Step 5, PN-B forwards the GET message to LN-B in Step 6 since LN-B happens to be awake. In Step 7, PN-B receives a CoAP ACK carrying an immediate response from LN-B. PN-B places the CoAP ACK in the payload of the RELOAD Tunnel response and routes it back to PN-A across the overlay in Step 8. PN-A forwards the CoAP ACK to LN-A in Step 9.

If LN-B was sleeping when PN-B receives the encapsulated CoAP GET in Step 5 of Figure 5, PN-B could either return cached information in the Tunnel response, or if no cached information was available, a CoAP ACK without an immediate response. When LN-B becomes available to produce a response, the response would be Tunneled across the overlay to PN-A inside a new RELOAD Tunnel request.

It is worth noting that although no observation relationship was established in Figure 5, the tunneling mechanism can be used even for observation relationships. In this case, the first tunneled CoAP method establishes the relationship. The actual notifications are sent in the payload of RELOAD Tunnel requests from PN-B to PN-A.

It is also worth noting that an observing node may not know in advance (i.e., before establishing the observation relationship) how frequent the notifications from the observed node will be. Therefore, the observing node can first establish a tunneled observation relationship. As the observing node receives tunneled notifications from the observed node, it can determine whether the frequency is high enough to justify the use of a dedicated connection. If this is the case, the observing node can switch from the tunnel to a dedicated connection.

6.4 Accessing CoAP resources from the web

When an HTTP client (e.g., an MCN) wishes to access a resource hosted by an LN, a GW node, having received an HTTP request from the MCN, maps from HTTP to CoAP and uses the CoAP usage for RELOAD for rendezvous. Figure 6 shows an example in which an MCN starts to observe a resource in a 6LoWPAN. In the example, a dedicated connection is used for the CoAP observation relationship. In Step 1 the MCN issues, using HTTP long polling [29], a hanging HTTP GET request to the GW node. A hanging GET is used since the MCN wishes to observe the CoAP resource, that is, receive notifications when new data becomes available. Having received the HTTP GET, the GW, which is acting as both a HTTP/CoAP proxy and a peer in the RELOAD overlay, sends in Step 2 a RELOAD Fetch (i.e., lookup) to the RELOAD overlay to learn the destination list of the LN hosting the resource specified in the HTTP GET. The Fetch is routed towards the node responsible for the resource-ID created by hashing the CoAP URI of the resource. In Step 3, the destination list of the LN is returned in a RELOAD Fetch 200 OK response. In Steps 4 and 5, a RELOAD Attach request is routed across the overlay to the PN behind which the LN is located in the overlay (the PN is the first entry on the destination list). The purpose of the Attach request is to establish a direct UDP connection between the GW and PN across NATs using ICE. The PN returns its ICE candidates in a RELOAD Attach 200 OK response in Steps 6 and 7. The ICE connectivity checks

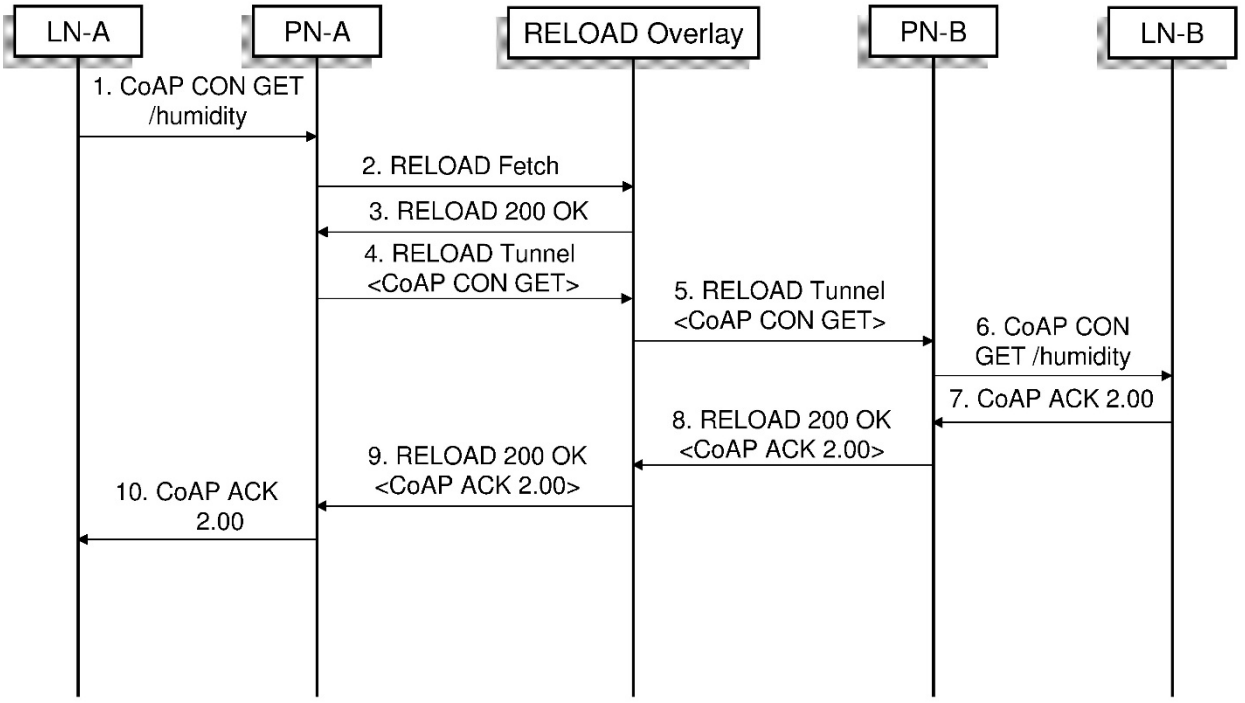

Figure 5 Use of Tunnel requests. The figure illustrates how CoAP messages can be sent tunneled in the payload of RELOAD messages across the overlay.

are performed in Step 8. In Step 9, ICE has established a direct UDP connection between the GW and the PN. Thus, the GW can send a CoAP GET to the PN using that connection. In Step 10, the PN, acting as a CoAP reverse proxy, sends the CoAP GET to the LN in the 6LoWPAN. The LN happens to be awake and returns an immediate response in a CoAP ACK in Steps 11 and 12 (if the LN was asleep, an ACK without immediate response would be returned). In Step 13, the GW answers the hanging HTTP GET and returns the requested information to the MCN. In Step 14, the MCN sends a new long poll request. Further notifications from the LN to the MCN are sent through the ICE-negiated connection between the PN and the GW. The next such notification is illustrated in Steps 15-17. A non-confirmable (NON) response is assumed in the figure.

6.5 Using the overlay as a cache

The CoAP usage for RELOAD also supports the use of the RELOAD overlay as a distributed cache for sensor data. As wireless sensors may typically be asleep for extended periods of time to maximize battery life, caching the most recent value of a sensor in the RELOAD overlay is a useful feature. The most straightforward way to achieve this is to store the value directly in the contact record of the sensor together with a timestamp. Besides the most recent value, the overlay can also be used to store and retrieve historical values (i.e., time series data) from a given sensor. In this case, the time series can be potentially divided to sub-series (to prevent the size of a single record from becoming too large), which are stored individually as resources in the overlay. Since the time series are stored as resources, they can be accessed using CoAP URIs.

LNs do not need to be aware of caching of sensor data in the overlay. As all the traffic to and from an LN goes through its PN, the decision to store a cached value in the overlay can be taken by the PN. As an example, when a PN forwards a CoAP response carrying a sensor reading from an LN to some other node, the PN can send a RELOAD Store request to cache the sensor reading in the overlay.

7 Simulations and use case

In this section, we will describe the simulation setup and the use case that we used to evaluate the proposed architecture.

7.1 Simulator

The simulations were run using our P2P simulator, which is an event-driven, message-level simulator. It

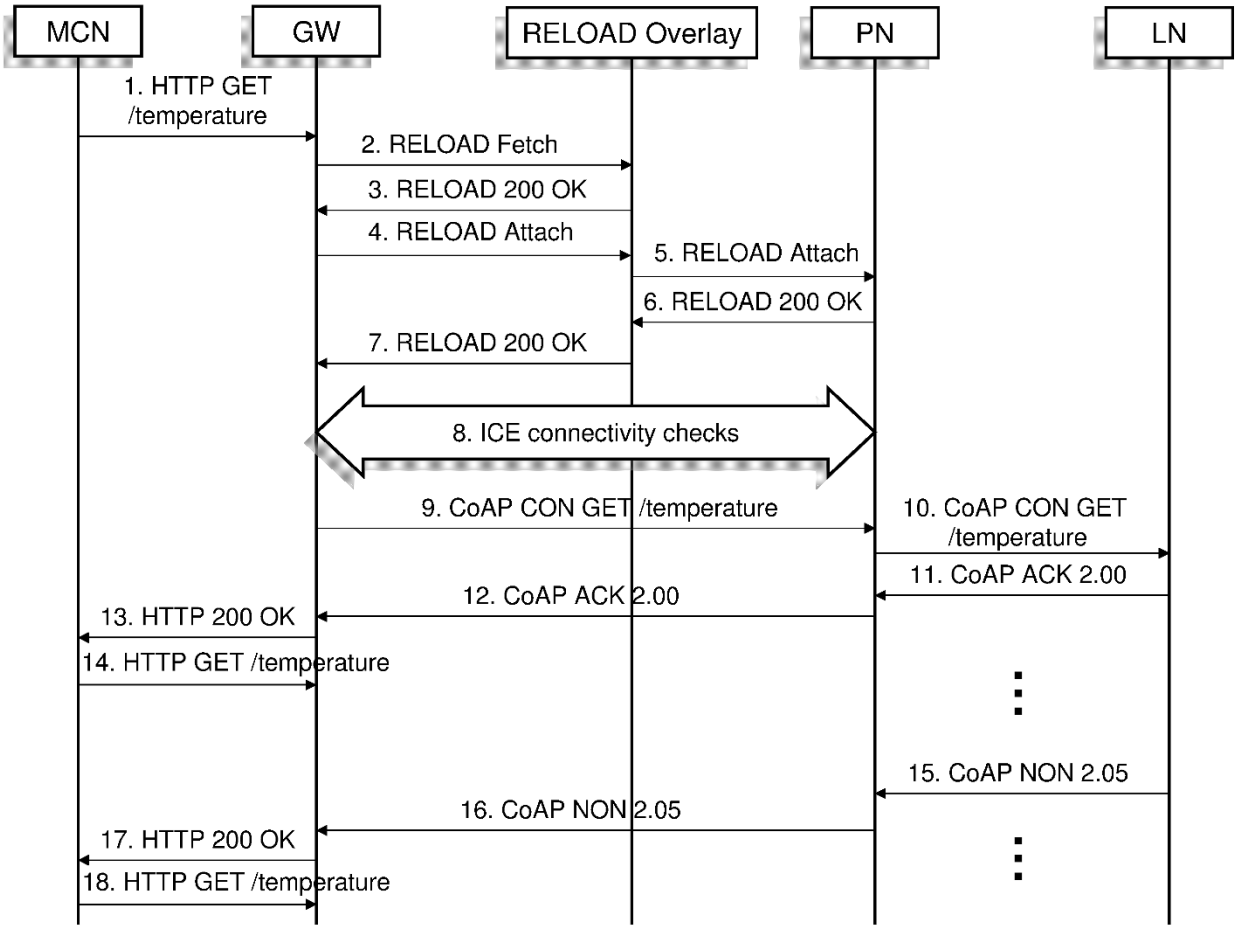

Figure 6 MCN getting information from LN. The figure shows how a Monitoring and Control Node (MCN) can access resources hosted by a Local Node (LN).

uses the same code base as our P2P prototype that we have used to run experiments in PlanetLab in previous study [22,30,31]. Since the simulator and the prototype use the same code base, new features (e.g., CoAP) added to the prototype are also immediately available in the simulator, and vice versa. We have previously used the simulator in [32]. We chose to use the simulator also in this article since we wanted to experiment with large-scale overlays created by only nodes using cellular radio access (that is, the PNs). Although our focus is on simulations, we will also evaluate the performance of our architecture through real-life experiments, which will be described in Section 8.3. The simulator is implemented in the Java programming language. It uses a predecessor of the RELOAD protocol called Peer-to-Peer Protocol (P2PP) [33] as the protocol between peers in the overlay. The current version of the RELOAD protocol [5], which is being standardized in the IETF, is based on P2PP (the P2PP proposal was merged with RELOAD during the IETF standardization process). All connections between peers are assumed to run over an unreliable transport. The Chord DHT [12] is used to organize the overlay. Chord was chosen since RELOAD specifies it as mandatory to implement [5].

The delay generator of the simulator uses a large set of delays that we collected by measuring delays in a 3G High Speed Downlink Packet Access (HSDPA) network. We measured delays between mobile devices, and between mobile devices and a central server. In the measurements, we used a large number of packet sizes ranging from 10 bytes to 1,400 bytes. The delay set was further extended by using results from our previous study of measuring ICE and RELOAD performance in mobile networks [21,22].

For this article, we extended the simulator by adding support for simulated LNs in WSNs that connect to the RELOAD overlay through a PN. We also implemented the CoAP protocol and integrated it to the simulator and the P2P prototype. All simulated PNs use ICE to establish connections for RELOAD and CoAP.

7.2 Use case

The use case we selected for evaluating our architecture is related to road traffic and road condition monitoring in the Finnish highway network. The Finnish highway network has nearly 10,000 kilometers of road. If monitoring units consisting of equipment to monitor traffic and weather conditions are deployed on highways for instance on the average every five kilometers, the resulting system has 2,000 monitoring units. With an average distance of one kilometer, there would be 10,000 monitoring units. We further assume that these units, which act as PNs, are equipped with 3G HSDPA modules. The monitoring units are coupled with variable speed limit displays and warning displays that are used to show information to the users of the road. Every monitoring unit acts as a PN for 1-100 wireless sensors (LNs) in a local WSN. The sensors are used for monitoring various conditions such as road surface temperature (to detect ice formation), air temperature, wind speed, wind direction, humidity, light, the amount of chemicals applied for prevention of freezing of water, water layer thickness, black ice, traffic volume, speed of vehicles (e.g., using a radar), and accidents (e.g., using video cameras coupled with computer vision). The data from the sensors can be used to adjust the current speed limit and show warnings to the users of the road on the warning displays. We assume that a given monitoring unit uses not only local information but information from sensors in nearby WSNs as input for showing warnings and for determining an appropriate speed limit to optimize the flow of traffic. For this, every monitoring unit is assumed to have on the average 10 outgoing CoAP observation relationships. The mean time between notifications is assumed to be either 1 min or 10 min.

In the simulations, we compare the performance of a RELOAD-based P2P architecture to a traditional Client/Server (C/S) architecture. In the C/S simulations, the RELOAD overlay was replaced by a central server. We also study the cost of using direct connections for CoAP observation relationships between sensors and their observers (Figure 4) to tunneling of CoAP messages across the overlay (Figure 5) or through a central server. Thus, we study four different scenarios. In Scenario 1, called *RELOAD-dedicated*, all the PNs are part of a RELOAD overlay and dedicated ICE-negotiated connections are used for CoAP observation relationships. In Scenario 2, called *RELOAD-tunnel*, no dedicated connections are set up for CoAP observation relationships. Instead, all notifications from sensors to observers are tunneled across the RELOAD overlay in the payload of RELOAD messages. In Scenario 3, called *C/S-dedicated*, there is no RELOAD overlay. Instead, a star topology in which all PNs communicate with a central server is used. However, there is still a P2P aspect present as dedicated CoAP observation relationships are established in a P2P manner directly between the sensors and their observers. In Scenario 4, *C/S-tunnel*, no dedicated connections are used for CoAP. Instead, all traffic, including CoAP notifications, are sent via the central server.

7.3 Traffic model and simulation parameters

We ran three sets of simulations whose setup is based on the use case described in Section 7.2. In the first set of simulations, the size of the overlay network was 2,000 PNs. The size of each WSN was 10 LNs, resulting in a total of 20,000 LNs. Each WSN had 10 outgoing CoAP observation relationships. Observed resources sent CoAP notifications on the average every 10 min. The duration of the simulated period was one hour. We assumed that all PNs are located behind P2P-friendly NATs using endpoint independent mapping and filtering behavior, meaning that relay servers were never needed. The reason why all the NATs were of the same type in the first set of simulations was to make it easier to compare delays between the four scenarios studied. We have studied the impact of different NAT types on delays in P2P networks in [22].

In the second set of simulations, the values of three parameters were modified compared to the first set of simulations. First, the size of the overlay network was increased to 10,000 PNs. Second, the number of LNs per WSN was varied between 1 and 100. Third, we assumed that a subset of the PNs are behind P2P-unfriendly NATs with address and port dependent mapping and filtering behavior. When two such PNs need to communicate with each other, a TURN server needs to relay all the traffic between the PNs. The types of NATs in mobile operator networks are studied in [23]. Based on these results, we assumed that 11.1% of connections between peers require the use of a relay. As a comparison, for Google Talk and Skype, the corresponding figures have been found to be 8% [34] and 9.6% [35], respectively. The purpose of varying the parameters was to experiment with higher traffic volumes than in the first set of simulations.

The third set of simulations was otherwise similar to the second one with the exception that the mean CoAP notification interval was reduced to 1 min. The goal was to investigate the effect of an even higher traffic volume caused by more frequent notifications.

The values of certain parameters were the same in all simulations. The Chord DHT's maintenance interval was set to 120s. With this maintenance rate, the overlay can handle a churn rate in which up to 250 peers depart or 500 peers join the overlay during an one hour period, which is more than enough for our use case, in which the overlay is expected to be fairly static. The size of the

Chord finger table (i.e., routing table) was set to eight peers following the recommendation in [12] to use on the order of $O(\log N)$ fingers (i.e., routing table entries). The size of Chord's successor and predecessor lists was set to three peers based on the minimum recommended in [5]. The ICE keepalive interval was set to 15s, which is the default value in ICE [13]. We used the same ICE stopping criteria as in [22].

We assume that when a PN receives a CoAP request destined to an LN, the PN always returns an immediate CoAP response in a CoAP ACK. Further, the PN always serves the CoAP request from a local cache instead of forwarding the request to the LN. This choice was made since in the simulations, we are only interested in comparing the delays associated with using a RELOAD overlay to those associated with using a central server. In the simulations, we are not interested in the delays within the WSNs as those are not affected by the fact whether the system uses a P2P or C/S architecture. Although we did not study the delays on the WSN side in the simulations, these delays were included in the end-to-end delay measurements of Section 8.3 that we carried out on real networks and hardware. The simulation parameters are summarized in Table 1 for the first set of simulations and in Table 2 for the second and third set of simulations. Table 2 shows only the parameters that are different from Table 1.

Table 1 Simulation parameters for the first set of simulations

Parameter	Value
Number of PNs	2,000
Number of LNs per WSN	10
Total number of LNs	20,000
Outgoing CoAP observations per WSN	10
CoAP notification interval (min)	10
Duration (s)	3,600
Chord finger pointers	8
Chord successors	3
Chord predecessors	3
Chord maintenance interval (s)	120
STUN keepalive interval (s)	15

Table 2 Parameters for the second and third set of simulations

Parameter	Value
Number of PNs	10,000
Number of LNs per WSN	1-100
Total number of LNs	10,000-10,00,000
CoAP notification interval (min)	1 or 10
Outgoing CoAP observations per LN	1
% of connections requiring a relay	11.1

8 Results

In this section, we will present the results of the simulations in Sections 8.1 and 8.2. In Section 8.3, we will present results of running a proof-of-concept prototype of our architecture on real networks and hardware.

8.1 Delays

In the first set of simulations, we focused on measuring the delays associated with establishing CoAP observation relationships. The delays of the four scenarios are shown in Figures 7 and 8. Figure 7 shows the average delay of establishing a CoAP observation relationship, whereas Figure 8 shows the average delay of individual CoAP transactions. The error bars in the figures represent 95% confidence intervals.

From Figure 7, we can observe that the delays of establishing a CoAP observation relationship using a dedicated ICE-negotiated connection are 28.1 s and 6.3 s in the RELOAD-dedicated and C/S-dedicated scenarios, respectively. When all CoAP messages are tunneled, the CoAP observation relationship establishment delays are 25.5s and 3.0 s for the RELOAD-tunnel and C/S-tunnel scenarios, respectively. As expected, the RELOAD delays are multiple times higher than the C/S delays. This is because of the additional delay associated with sending RELOAD Fetch, Attach, and Tunnel messages over multiple hops across the overlay. Each hop involves sending the message twice over the 3G wireless radio interface (i.e., in the sender's and receiver's wireless access networks).

From the figure, we can observe that the reason why dedicated connections are more expensive for both RELOAD and C/S scenarios than the use of RELOAD Tunnel requests to establish the observation relationships is due to the ICE negotiations, which take roughly 3.2s for both RELOAD and C/S (the difference in delays between RELOAD and C/S scenarios is not statistically significant).

Figure 8 shows the delays of subsequent CoAP transactions sent after the observation relationship has been established. As expected, the delays are equal, roughly 1.1 s for both the RELOAD-dedicated and C/S-dedicated scenarios. The delays of tunneling CoAP messages across the overlay or via the central server are 11.7 s and 1.8 s for the RELOAD-tunnel and C/S-tunnel scenarios, respectively. Thus, the tunneled scenarios have clearly higher cost than the scenarios using dedicated connections. Especially the cost of tunneling CoAP across the RELOAD overlay is so high that it may not be feasible in practice if observers require real-time or near-real time information from the sensors.

Although the delay associated with establishing the CoAP observation relationship is 4.5 times higher for

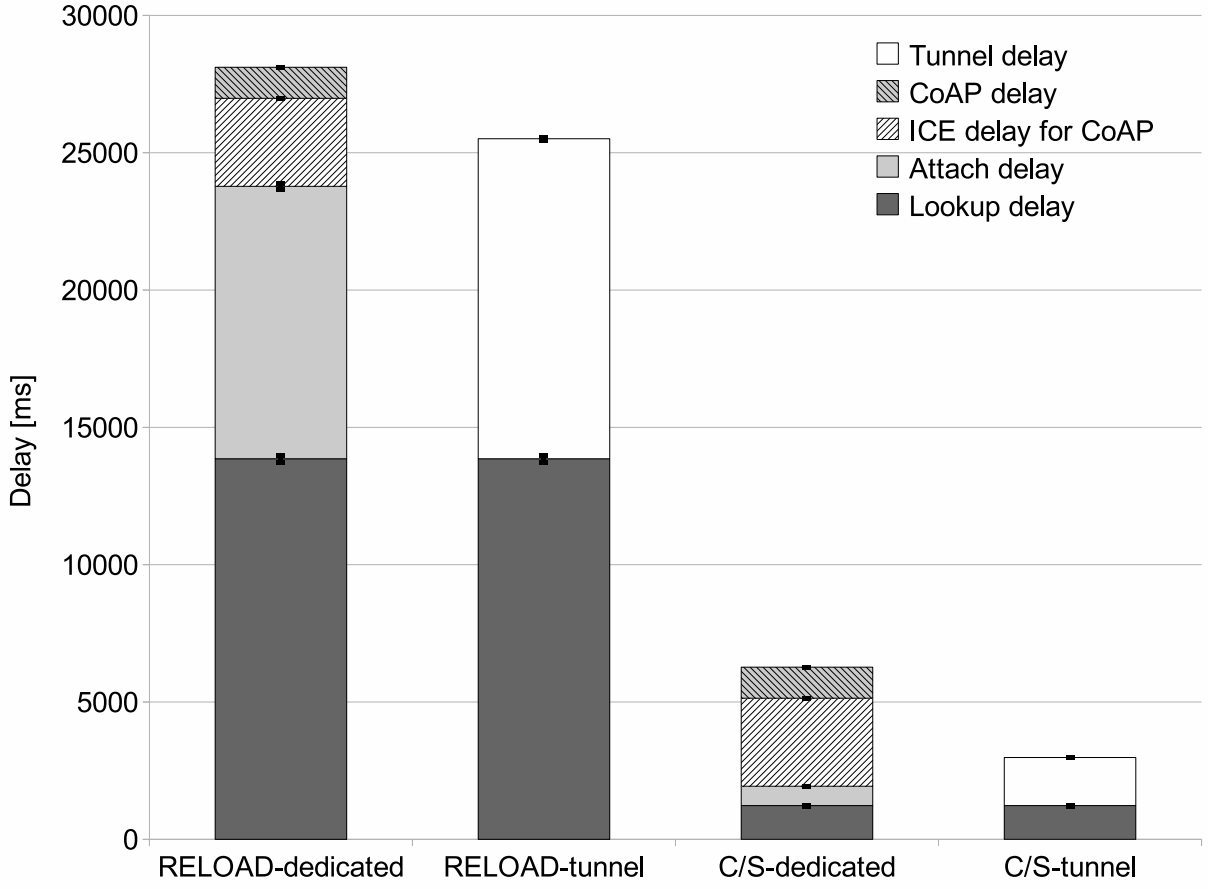

Figure 7 Delay of establishing CoAP observation relationship. The figure shows the average delay of establishing a CoAP observation relationship. The figure compares four scenarios: (1) RELOAD overlay is used together with dedicated connections for CoAP, (2) RELOAD overlay is used and CoAP signaling is sent tunneled across the overlay, (3) client/server architecture is used and dedicated connections are used for CoAP, and (4) client/server architecture is used and CoAP messages are sent via a central server.

the RELOAD-dedicated than for the C/S-dedicated scenario, it is worth noting that this delay occurs only once when establishing the relationship. In the RELOAD-dedicated scenario, after this one-time cost, subsequent CoAP transactions experience no extra delay compared to the C/S-dedicated case. Thus, if this one-time additional cost can be tolerated, as one would expect in a typical use case, the use of a RELOAD overlay is no more expensive than the use of a central server when it comes to CoAP delays.

8.2 Traffic load

8.2.1 First set of simulations

Since all of the PNs use cellular radio access, it is interesting to compare the total traffic loads generated in our four scenarios. The traffic load should be minimized to minimize the load on the cellular access network and also for energy efficiency reasons. We will first describe the results for traffic load in the first set of simulations. The results of the second and third set of simulations are described in the sections below.

The total amount of application protocol (i.e., CoAP, RELOAD, and STUN) traffic exchanged in the overlay during the one-hour period in the first set of simulations is shown in Figure 9 for our four scenarios. In the RELOAD-dedicated scenario, the traffic consists of CoAP messages, STUN keepalive traffic for RELOAD and CoAP, and RELOAD overlay maintenance traffic. In the RELOAD-tunnel scenario, the traffic consists of RELOAD maintenance traffic, RELOAD application traffic (i.e., Tunnel requests), and STUN keepalives for RELOAD. In the C/S-dedicated scenario, the traffic consists of STUN keepalives for CoAP and RELOAD (we used RELOAD messages for client/server signaling between the PNs and the central server), and CoAP messages. In the C/S-tunnel scenario, the traffic consists

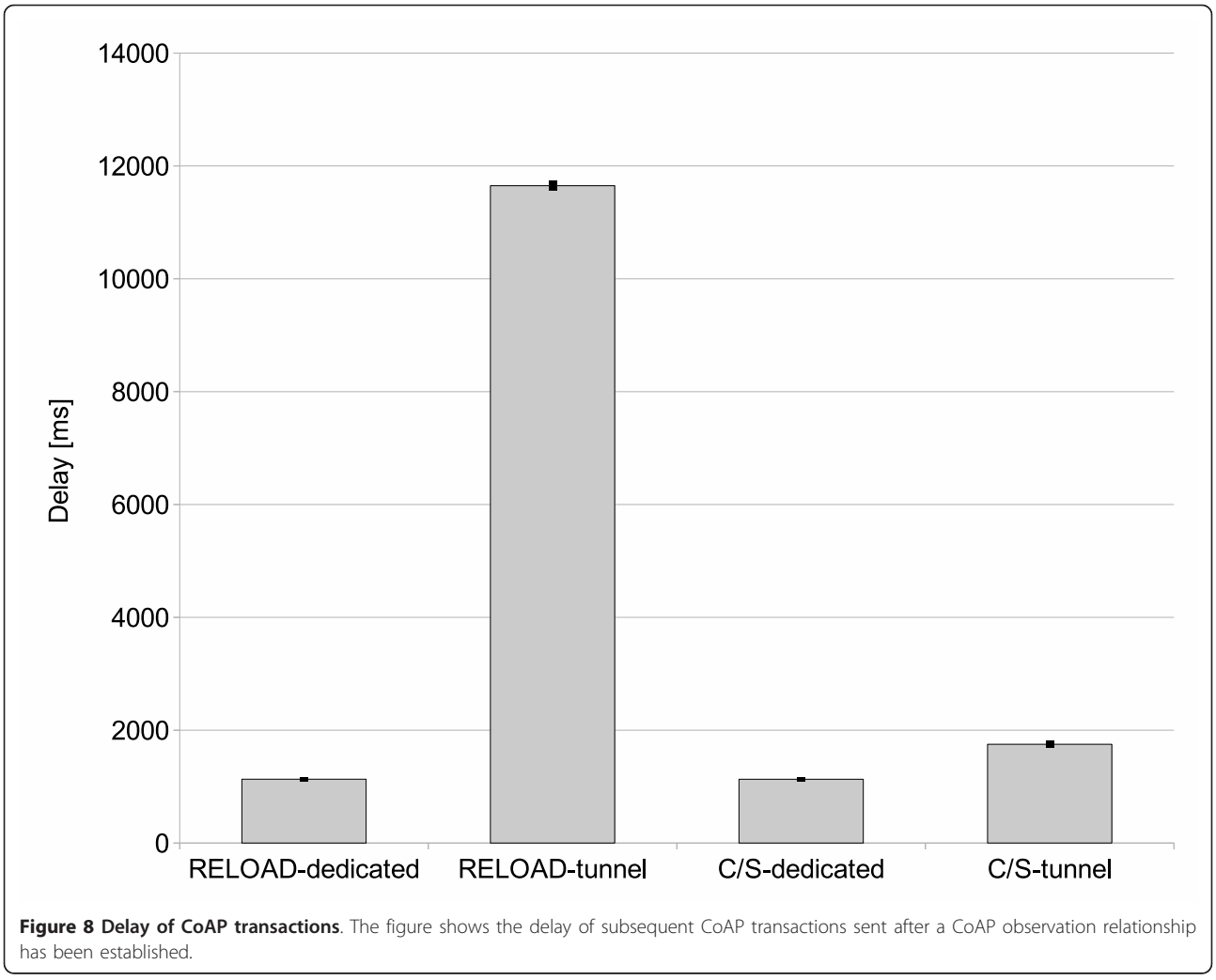

Figure 8 Delay of CoAP transactions. The figure shows the delay of subsequent CoAP transactions sent after a CoAP observation relationship has been established.

only of STUN keepalives and CoAP messages (CoAP messages were encapsulated in RELOAD Tunnel messages).

From the figure, we can observe that in the RELOAD-dedicated scenario, the total traffic during the one-hour period is 938 MB, whereas in the C/S-dedicated scenario it is 365 MB. The RELOAD-tunnel scenario generates 750MB of traffic, whereas C/S-tunnel generates only 124 MB. The higher cost of the RELOAD scenarios is explained especially by the RELOAD overlay maintenance traffic and high amount of STUN traffic required to keep the RELOAD connections between PNs alive (every PN maintains connections to all peers in the routing table, whose size is 8 fingers, 3 successors, and 3 predecessors as described in Table 1). In all of the scenarios, a large part of the total traffic is STUN keeplives. The percentages are 71%, 43%, 96%, and 53% for the RELOAD-dedicated, RELOAD-tunnel, C/S-dedicated, and C/S-tunnel scenarios, respectively. For the RELOAD-dedicated and C/S-dedicated scenarios, the largest source of traffic are STUN keepalives for CoAP connections. The percentage of STUN keepalives for CoAP out of total traffic are 40% and 89% for the RELOAD-dedicated and C/S-dedicated scenarios, respectively.

Thus, we can conclude that, as expected, with the traffic model described in Table 1, the C/S scenarios generate considerably less total traffic than the RELOAD scenarios. However, the main difference is of course that in the C/S scenario, the central server needs to handle either all (C/S-tunnel) or a part (C/S-dedicated) of the total traffic. Therefore, it is interesting to study also the incoming traffic that the central server needs to handle. We will do this in the second set of simulations described below.

8.2.2 Second set of simulations

In the second set of simulations, we compared the traffic load of our four different scenarios in more challenging conditions (that were described in Section 7.3).

The total amount of traffic exchanged in each of the four scenarios in the second set of simulations is shown

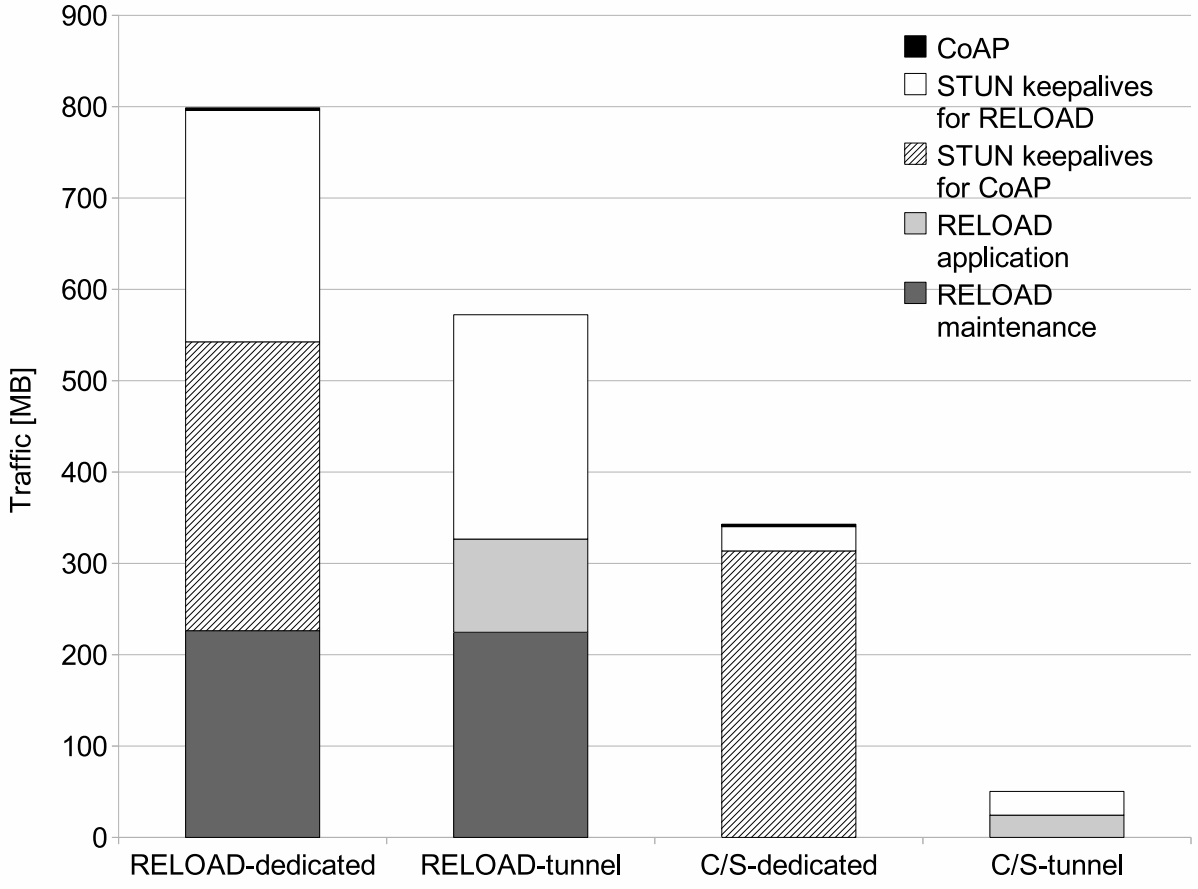

Figure 9 Total traffic, 2,000 proxy nodes. The figure shows the total amount of application protocol traffic exchanged in the system during the measurement period when using 2,000 proxy nodes and a 10 min CoAP notification interval. The proxy nodes were assumed to be located behind P2P-friendly NATs.

in Figure 10. The *y*-axis in the figure uses the logarithmic scale. From the figure, we can see that the C/S-tunnel scenario still has the lowest amount of total traffic. Unlike in Figure 9, the RELOAD-tunnel scenario has now the highest total amount of traffic (except for the case when there is only one LN per WSN). The amount of traffic in the C/S-dedicated scenario starts to approach and eventually becomes nearly equal to the traffic in the RELOAD-dedicated scenario as the number of LNs per WSN increases. This is because CoAP messages and STUN keepalives for CoAP start to dominate and thus the additional cost of using RELOAD over C/S becomes negligible when looking at the total traffic levels.

Figure 11 shows the number of incoming Mbit/s for the central server in the C/S scenarios and for an average peer in the RELOAD scenarios. The *y*-axis in the figure uses the logarithmic scale. From the figure, we can see that in the RELOAD scenarios, the PNs are not especially loaded. However, in the C/S scenarios, the load of the central server starts to grow rapidly. For instance, when there are 10 LNs per WSN, the server needs to already be able to handle an incoming traffic load of 3.6 and 11.1 Mbits/s in the C/S-dedicated and C/S-tunnel scenarios, respectively. Thus, we can conclude that the scalability of the RELOAD scenarios appears to be much better than the scalability of the C/S architecture.

8.2.3 Third set of simulations

In our third and final set of simulations, we decreased the mean interval of CoAP notifications to 60 s, while keeping the values of all the other parameters identical compared to the second set of simulations. The resulting total traffic in our four scenarios is shown in Figure 12. From the figure, we can observe that when the frequency of CoAP notifications is high, both the RELOAD-tunnel and C/S-tunnel scenarios become clearly more expensive than the RELOAD-dedicated and

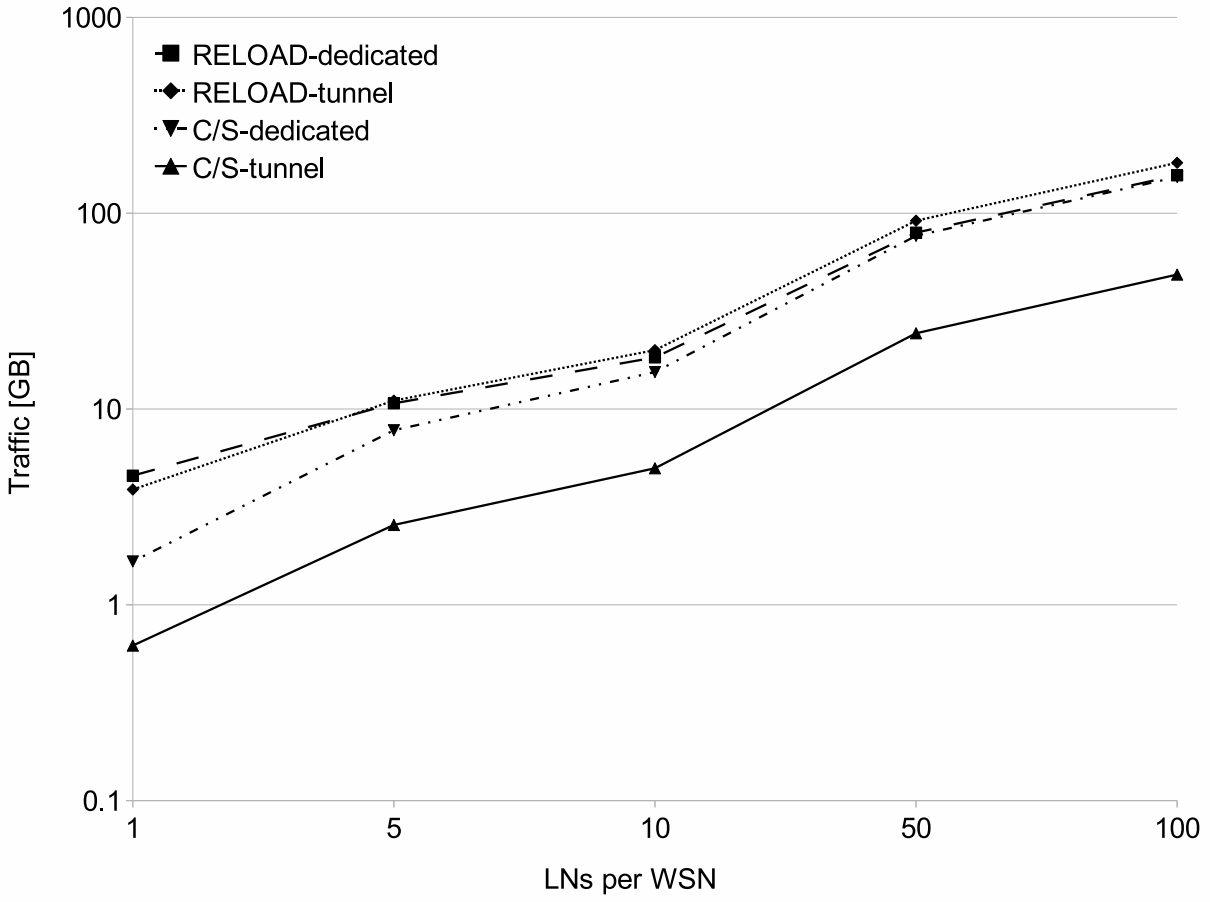

Figure 10 Total traffic, 10,000 proxy nodes. The figure shows the total amount of application protocol traffic exchanged in the system during the measurement period when using 10,000 Proxy Nodes (PNs), 1-100 Local Nodes (LNs) per PN, and a 10 min CoAP notification interval. A subset of the PNs was assumed to be located behind P2P-unfriendly NATs.

C/S-dedicated scenarios in terms of total traffic. Further, when looking at the amount of incoming Mbit/s per average peer (RELOAD scenarios) or the central server (C/S scenarios) shown in Figure 13, we can see that the RELOAD-dedicated scenario clearly outperforms the other scenarios. In the worst case (100 LNs per WSN), the average Mbit/s received by an average peer or the central server is 0.04 Mbit/s, 0.40 Mbit/s, 42Mbit/s, and 1078 Mbit/s for the RELOAD-dedicated, RELOAD-tunnel, C/S-dedicated, and C/S-tunnel scenarios, respectively. Thus, we can again see that the RELOAD scenarios (especially RELOAD-dedicated) scale very well, whereas in the C/S scenarios, the traffic load can become very high for the central servers.

8.3 Evaluation of the system on real network and hardware

We are working on a proof-of-concept prototype of our wide area sensor and actuator networking architecture. In the prototype, we use small Gumstix Overo Earth COM[a] single-board computers having a 600 MHz ARM Cortex-A8 CPU as the PNs. The single-board computers run an embedded Linux operating system. We have equipped the single-board computers with a 3G dongle and a Libelium Waspmote[b] ZigBee Gateway dongle, both of which are connected via USB. The 3G dongle takes care of wide area (i.e., Internet) connectivity. The ZigBee Gateway dongle is used for communication towards a ZigBee WSN. We use Libelium Waspmote ZigBee sensors as legacy LNs. The prototype uses the same code base as our simulator. This is enabled by an abstraction layer hiding away the fact whether a real or simulated networking layer is used. Since our prototype is Java-based, we run the CACAO Java Virtual Machine, [c] which supports ARM processors, on the single-board computers. We also developed some extra modules for the prototype. These include for instance a module taking care of interworking between CoAP/UDP/IP and the

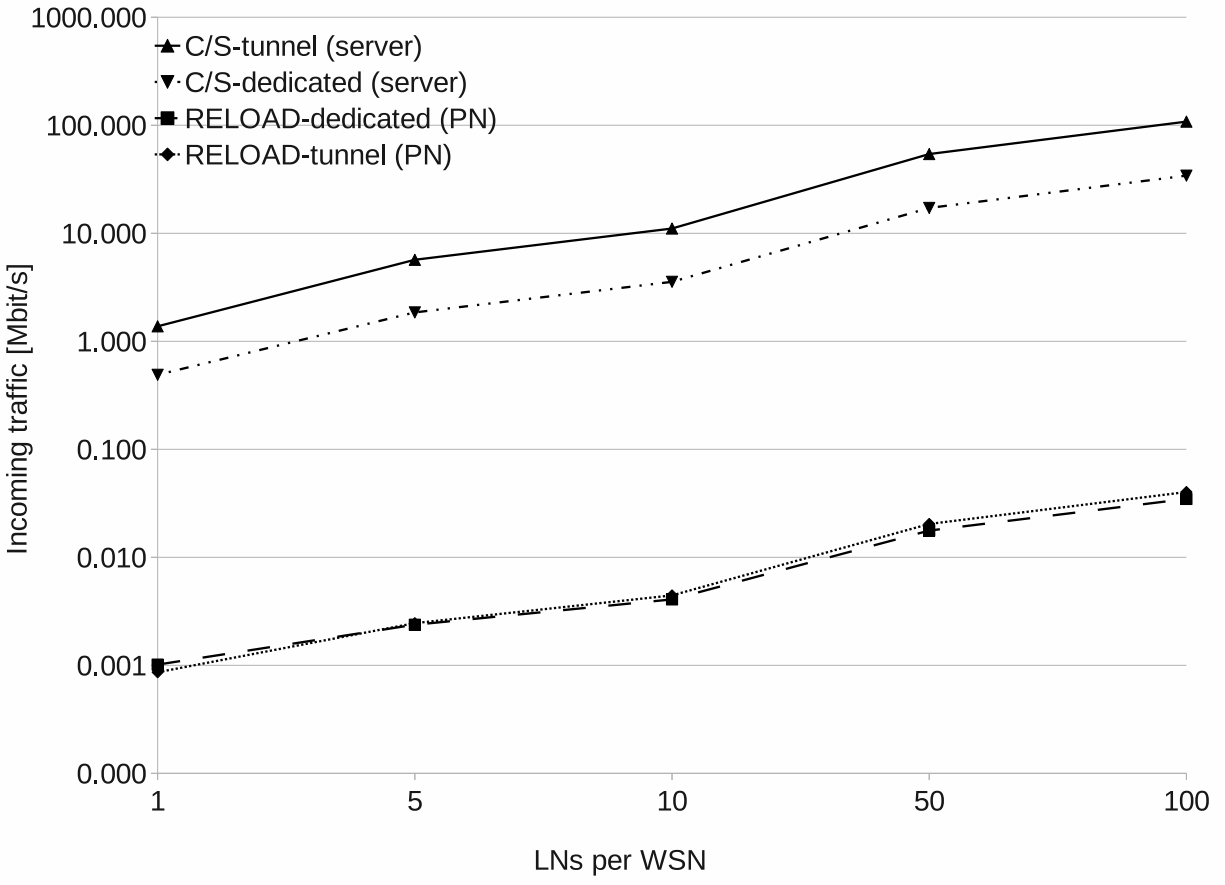

Figure 11 Incoming traffic per second, 10,000 proxy nodes. The figure shows the number of incoming Mbit/s for the central server in the C/S scenarios and for an average peer in the RELOAD scenarios when using 10,000 Proxy Nodes (PNs), 1-100 Local Nodes (LNs) per PN, and a 10 min CoAP notification interval. A subset of the PNs was assumed to be located behind P2P-unfriendly NATs.

ZigBee protocol stack. Finally, we use a third-party ZigBee API library called xbee-api[d] for interfacing the prototype with the ZigBee gateway device.

We ran a series of extra measurements using our proof-of-concept prototype. All these measurements used the RELOAD-dedicated scenario. In the measurements, two PNs, PN-A, and PN-B, were acting as clients in a 1,000-node RELOAD overlay network running in PlanetLab. We chose to use a PlanetLab overlay rather than an overlay created only by PNs to be able to experiment with a reasonably large number of nodes (we had only a limited number of the PN hardware available). The PlanetLab nodes were running the same Java Standard Edition version of our prototype as the single-board computers. Both PNs were located behind NATs using endpoint independent mapping and filtering behavior. In the measurements, we focused on measuring the communication delays associated with one LN, LN-A, starting to observe a CoAP resource of another LN, LN-B, located behind a different PN. We measured separately the delays between each LN and its PN and the delays between the two PNs. Since the overlay was running in PlanetLab, only the first hop (PN to PlanetLab node) and the last hop (PlanetLab node to PN) in the overlay went over cellular radio. In the measurements, the LNs and PNs were located in the same room with no obstacles between them.

The results of the measurements are shown in Table 3. The table contains the average delay calculated over 100 measurements for each component of the delay. The delay of ICE candidate gathering at LN-A is shown separately in the table. In the table, the value in the parentheses is the standard deviation. The results look slightly different compared to the delay results in our simulations due to the fact that in the measurements, only the first and last hops in the RELOAD overlay went over the 3G radio interface; all the intermediate routing hops occurred between PlanetLab nodes. We

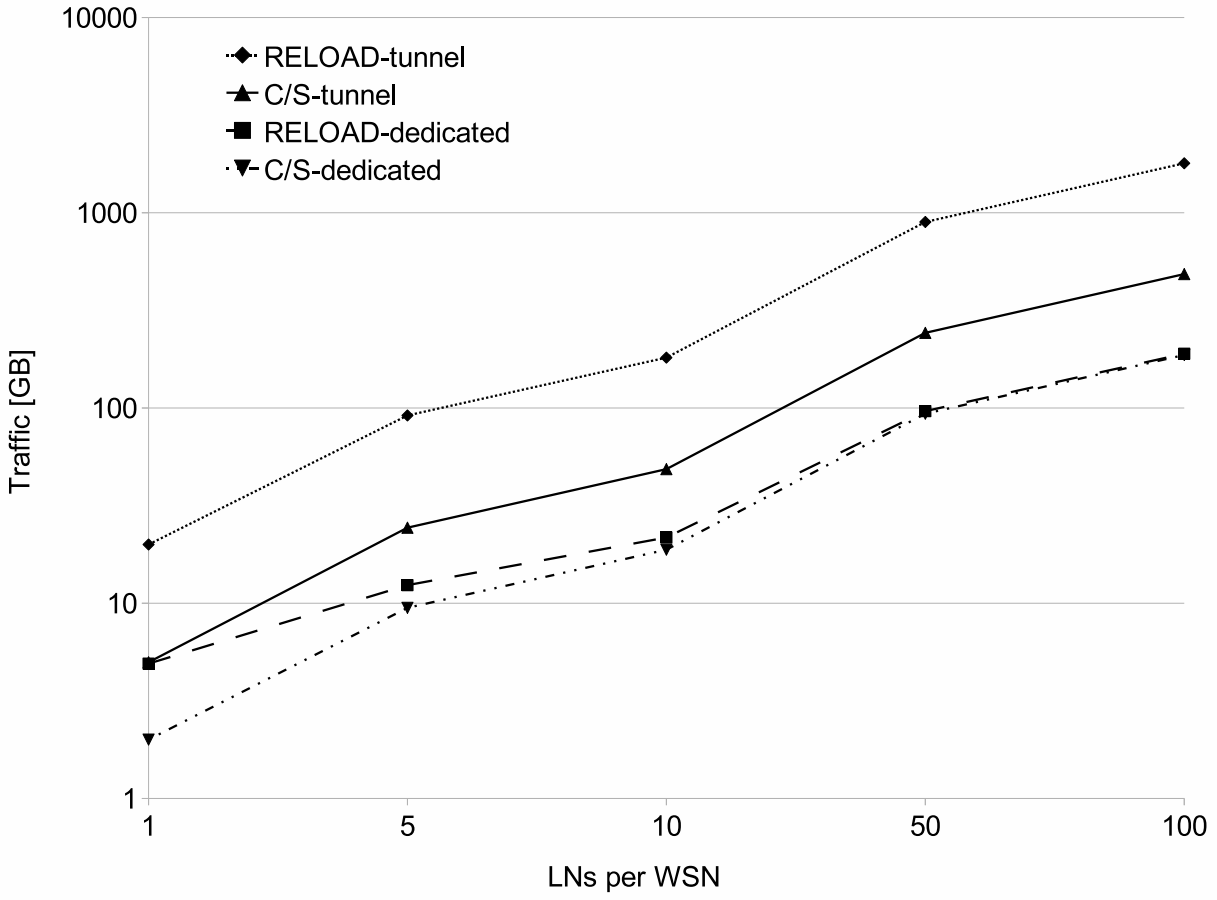

Figure 12 Total traffic, 10,000 proxy nodes, 60-s notification interval. The figure shows the total amount of application protocol traffic exchanged in the system during the measurement period when using 10,000 Proxy Nodes (PNs), 1-100 Local Nodes (LNs) per PN, and a 60s CoAP notification interval. A subset of the PNs was assumed to be located behind P2P-unfriendly NATs.

can see that the total delay from the moment that LN-A initiates a CoAP message to the moment it receives the CoAP response from LN-B is on the average 12.9s. Communication over ZigBee represents only 10% of the total delay. The largest component of the delay is the Attach procedure. The large Attach delay, especially compared to the considerably lower Lookup delay, is explained by two factors. First, the Attach request and response go twice over the 3G radio interface (in PN-A's and PN-B's access networks). In contrast, the Lookup request is answered by a PlanetLab node and thus goes over 3G only on the sending PN's side. In general, all delay components that require messaging over the 3G network on both the A and B side, are high. These include the Attach delay, ICE negotiation delay, and PN-to-PN CoAP delay. The second reason for the high Attach delay is that it also includes ICE candidate gathering at PN-B. The main additional finding from these measurements compared to the simulations is that communication between the LNs and PNs represents only a minor part of the end-to-end delay.

9 Conclusions

In this article, we proposed a new architecture for wide area sensor and actuator networking. The architecture uses the CoAP and RELOAD protocols to provide a P2P federation of geographically distributed WSNs. For this, we defined a CoAP usage for RELOAD. The usage provides a distributed rendezvous, storage, data caching, and NAT traversal service for CoAP endpoints. One of the major benefits of the architecture is that it is completely decentralized, that is, not dependent on central application servers, central resource directories, and centralized services like DNS-SD, whose use has not been defined for CoAP yet. Other advantages include scalability, self-organization, robustness, cost-efficiency (both low capex and opex), and web integration.

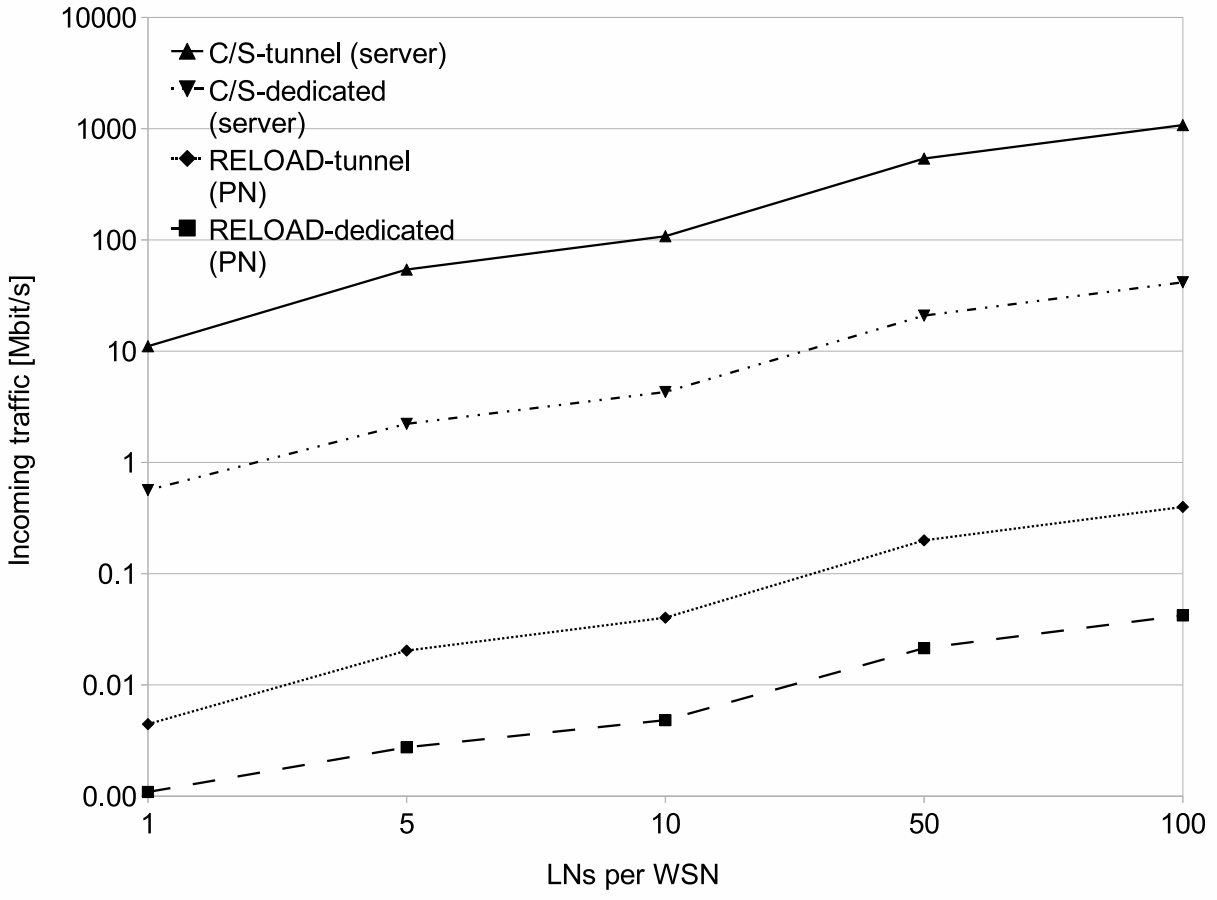

Figure 13 Incoming traffic per second, 10,000 proxy nodes, 60-s notification interval. The figure shows the number of incoming Mbit/s for the central server in the C/S scenarios and for an average peer in the RELOAD scenarios when using 10,000 Proxy Nodes (PNs), 1-100 Local Nodes (LNs) per PN, and a 60s CoAP notification interval. A subset of the PNs was assumed to be located behind P2P-unfriendly NATs.

We implemented the architecture and evaluated it through simulations, comparing its performance to that of a client/server system. A client/server system generates less traffic when the network is small and the volume of inter-device communication (i.e., number of CoAP observation relationships and the frequency of CoAP notifications) is low. A client/server system also puts a lower load on the Radio Access Network (RAN) since in a P2P system, each message goes through the RAN twice, both on the side of the sending and receiving peer. Further, the server of a client/server system is typically located in a high-speed network. However, a client/server system scales very badly as the network becomes larger or the volume of inter-device communication increases. The communication delays of client/server and decentralized systems are on the same level when using dedicated connections for CoAP observation relationships. However, when CoAP messages are tunneled, which is an appropriate strategy when the volume of inter-device communication is low, the delays associated with the decentralized architecture are higher.

Table 3 Measurements in PlanetLab

Component	Delay (ms)
LN-to-PN CoAP delay over ZigBee	1316 (189)
Lookup delay	1786 (189)
ICE candidate gathering at LN-A	1058 (268)
Attach delay	3462 (1284)
ICE negotiations	3200 (372)
PN-to-PN CoAP delay	2149 (547)
Total delay	12971 (2108)

Thus, which architecture to recommend depends on various factors. The general advantages associated with P2P architectures include self-organization, low capex and opex, robustness and scalability. If the frequency of CoAP messaging is low, the size of the system small, the

number of CoAP observation relationships low or if there is a need to minimize RAN load, and the above-mentioned advantages of P2P systems are less important, then a client/server architecture is the best choice. However, if the advantages of P2P architectures are important for the use case or as the frequency of CoAP messaging, the size of the system, or the number of observation relationships grows, a P2P architecture becomes quickly more recommendable. Based on these findings, the types of sensor and actuator networks that benefit most from our architecture are large-scale networks having from moderate to high levels of inter-device communication.

We also implemented a proof-of-concept prototype of our our wide area sensor and actuator network and tested the prototype on real networks and hardware. In these measurements, we focused on observing the end-to-end communication delay between two sensor and actuator devices. Among other things, we discovered that the dominant part of the end-to-end delay associated with one device accessing a resource of another device is communication between the proxy nodes across the overlay. We aim to further analyze the performance of our proof-of-concept prototype in future study.

Endnotes

[a]http://www.gumstix.com. [b]http://www.libelium.com/products/waspmote.

[c]http://www.complang.tuwien.ac.at/cacaojvm/. [d]http://code.google.com/p/xbee-api/.

Abbreviations

2G: Second Generation; 3G: Third Generation; 3GPP: Third Generation Partnership Project; 6LBR 6LoWPAN: Border Router; 6LoWPAN: IPv6 over LoW Power wireless Area Networks; C/S: Client/Server; CoAP: Constrained Application Protocol; CPU: Central Processing Unit; CSDS: CoAP Server Discovery Server; CSN: Chord for Sensor Networks; DHCPv6: Dynamic Host Configuration Protocol for IPv6; DHT: Distributed Hash Table; DNS: Domain Name System; DNS SD DNS: Service Discovery; GHT: Geographic Hash Table; GW: Gateway, HTTP: Hypertext Transfer Protocol; HSDPA: High Speed Downlink Packet Access ICE Interactive Connectivity Establishment; ID: Identifier; IETF: Internet Engineering Task Force; IP: Internet Protocol; IPv6: Internet Protocol version 6; LoWPAN: Low Power Wireless Personal Area Network; LN: Local Node; MCN: Monitoring and Control Node; NAT: Network Address Translator; P2P: Peer-to-Peer; P2PSIP: Peer-to-Peer Session Initiation Protocol; PN: Proxy Node; RAM: Random Access Memory; RAN: Radio Access Network; RELOAD: REsource LOcation And Discovery; REST: Representational State Transfer; SHA-1: Secure Hash Algorithm One; STUN: Session Traversal Utilities for NAT; TURN: Traversal Using Relays around NAT; UDP: User Datagram Protocol; URI: Uniform Resource Identifier; VCP: Virtual Cord Protocol; WN: Wide area Node; WSN: Wireless Sensor Network.

Competing interests

The authors declare that they have no competing interests.

Received: 14 June 2011 Accepted: 27 March 2012
Published: 27 March 2012

References

1. More than 50 billion connected devices - taking connected devices to mass market and profitability. White paper, Ericsson http://www.ericsson.com/res/docs/whitepapers/wp-50-billions.pdf (2011). [Available online (12 pages)]
2. G Santucci, The Internet of Things: Between the Revolution of the Internet and the Metamorphosis of Objects, in *Vision and Challenges for Realising the Internet of Things*, ed. by H Sundmaeker, P Guillemin, P Friess, S Woelffl European Commission, pp. 11–24 (2010)
3. Device Connectivity Unlocks Value. White paper, Ericsson http://www.ericsson.com/res/docs/whitepapers/wp-device-connectivity.pdf (2011). [Available online (13 pages)]
4. Z Shelby, K Hartke, C Bormann, B Frank, Constrained Application Protocol (CoAP), draft-ietf-core-coap-08. Internet draft, IETF 2011 (in progress)
5. C Jennings, B Lowekamp, E Rescorla, S Baset, H Schulzrinne, REsource LOcation And Discovery (RELOAD) Base Protocol, draft-ietf-p2psip-base-20. Internet draft, IETF 2012 (in progress)
6. K Hartke, Z Shelby, Observing Resources in CoAP, draft-ietf-core-observe-04. Internet draft, IETF 2012 (in progress)
7. Z Shelby, CoRE Link Format, draft-ietf-core-link-format-11. Internet draft, IETF 2012 (in progress)
8. Z Shelby, S Krco, CoRE Resource Directory, draft-shelby-core-resource-directory-02. Internet draft, IETF 2011 (in progress)
9. P van der Stok, K Lynn, CoAP Utilization for Building Control, draft-vanderstok-core-bc-05. Internet draft, IETF 2011 (in progress)
10. G Montenegro, N Kushalnagar, J Hui, D Culler, Transmission of IPv6 Packets over IEEE 802.15.4 Networks. *RFC 4944*. IETF 2007
11. Z Shelby, M Garrison Stuber, D Sturek, B Frank, R Kelsey, CoAP Requirements and Features, draft-shelby-core-coap-req-04. Internet draft, IETF 2011 (in progress)
12. I Stoica, R Morris, D Liben-Nowell, DR Karger, MF Kaashoek, F Dabek, H Balakrishnan, Chord: a scalable peer-to-peer lookup protocol for internet applications. **11**, 17–32 (2003) IEEE/ACM Trans. Netw
13. J Rosenberg, Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols. *RFC 5245*. IETF 2010
14. C Jennings, B Lowekamp, E Rescorla, S Baset, H Schulzrinne, A SIP Usage for RELOAD, draft-ietf-p2psip-sip-07. Internet draft, IETF 2012 (in progress)
15. J Rosenberg, R Mahy, P Matthews, D Wing, Session Traversal Utilities for NAT (STUN). *RFC 5389*. IETF 2009
16. R Mahy, P Matthews, J Rosenberg, Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). *RFC 5766*. IETF 2010
17. AAB Al-Mamou, H Labiod, ScatterPastry: An Overlay Routing Using a DHT over Wireless Sensor Networks, in *Proceedings of The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007), IEEE Computer Society*, Jeju Island, Korea, pp. 274–279 (2007)
18. A Awad, C Sommer, R German, F Dressler, Virtual cord protocol (VCP): a flexible DHT-like routing service for sensor networks, in *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008)*, Atlanta, USA, pp. 133–142 (2008)
19. S Ratnasamy, B Karp, S Shenker, D Estrin, R Govindan, L Yin, F Yu, Data-centric storage in sensornets with GHT, a geographic hash table. Mob Netw Appl. **8**, 427–442http://dx.doi.org/10.1023/A:1024591915518 (2003). doi:10.1023/A:1024591915518
20. M Ali, ZA Uzmi, CSN: A network protocol for serving dynamic queries in large-scale wireless sensor networks, in *Proceedings of the Second Annual Conference on Communication Networks and Services Research*, IEEE Computer Society, Washington, DC, USA, pp. 165–174 (2004) http://portal.acm.org/citation.cfm?id=998669.998893
21. J Mäenpää, J Jimenez, Performance of REsource LOcation and Discovery (RELOAD) on mobile phones, in *Proceedings of the 2010 IEEE Wireless Communications and Networking Conference (IEEE WCNC)*, Sydney, Australia, pp. 1–6 (2010)
22. J Mäenpää, V Andersson, A Keränen, G Camarillo, Impact of network address translator traversal on delays in peer-to-peer session initiation, in *Proceedings of the 2010 IEEE Global Telecommunications Conference (IEEE GLOBECOM)*, Miami, USA, pp. 1–6 (2010)
23. L Mäkinen, JK Nurminen, Measurements on the feasibility of TCP NAT traversal in cellular networks, in *Proceedings of the 4th EURO-NGI Conference*

on Next Generation Internet Networks (NGI 2008), IEEE, Krakow, Poland, pp. 261–267 (2008)
24. E Kim, D Kaspar, C Gomez, C Bormann, Problem Statement and Requirements for 6LoWPAN Routing, draft-ietf-6lowpan-routing-requirements-10. Internet draft, IETF 2011 (in progress)
25. A Castellani, S Loreto, A Rahman, T Fossati, E Dijk, Best practices for http-CoAP mapping implementation, draft-castellani-core-http-mapping-02. Internet draft, IETF 2011 (in progress)
26. S Cheshire, M Krochmal, DNS-Based Service Discovery, draft-cheshire-dnsext-dns-sd-11. Internet draft, IETF 2011 (in progress)
27. D Guinard, M Fischer, V Trifa, Sharing using social networks in a composable Web of Things, in *Proceedings of the 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, Mannheim, Germany, pp. 702–707 (2010)
28. C Bormann, CoRE Simple Server Discovery, draft-bormann-core-simple-server-discovery-00. Internet draft, IETF 2011 (in progress)
29. S Loreto, P Saint-Andre, S Salsano, G Wilkins, Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional http. RFC 6202. IETF 2011
30. J Mäenpää, G Camarillo, Study on maintenance operations in a peer-to-peer session initiation protocol overlay network, in *Proceedings of the 2009 23rd IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2009)*, Rome, Italy, pp. 1–9 (2009)
31. J Mäenpää, G Camarillo, Analysis of delays in a peer-to-peer session initiation protocol overlay network, in *Proceedings of the 2010 7th IEEE Consumer Communications and Networking Conference (CCNC 2010)*, Las Vegas, USA, pp. 1–6 (2010)
32. J Mäenpää, G Camarillo, Estimating operating conditions in a session initiation protocol overlay network, in *Proceedings of the 2010 24th IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2010)*, Atlanta, USA, pp. 1–8 (2010)
33. S Baset, H Schulzrinne, M Matuszewski, Peer-to-Peer protocol (P2PP). Internet draft, IETF 2007 (in progress)
34. Google Talk for Developers http://code.google.com/apis/talk/libjingle/ (2011)
35. S Guha, N Daswani, R Jain, An experimental study of the Skype peer-to-peer VoIP system, in *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS 2006)*, Santa Barbara, CA, USA, pp. 1–6 (2006)

doi:10.1186/1687-1499-2012-121
Cite this article as: Mäenpää *et al.*: Using RELOAD and CoAP for wide area sensor and actuator networking. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:121.